Práctica 4:

Robustez y Modularidad Manejo de Errores y Sub-Flujos de Trabajo

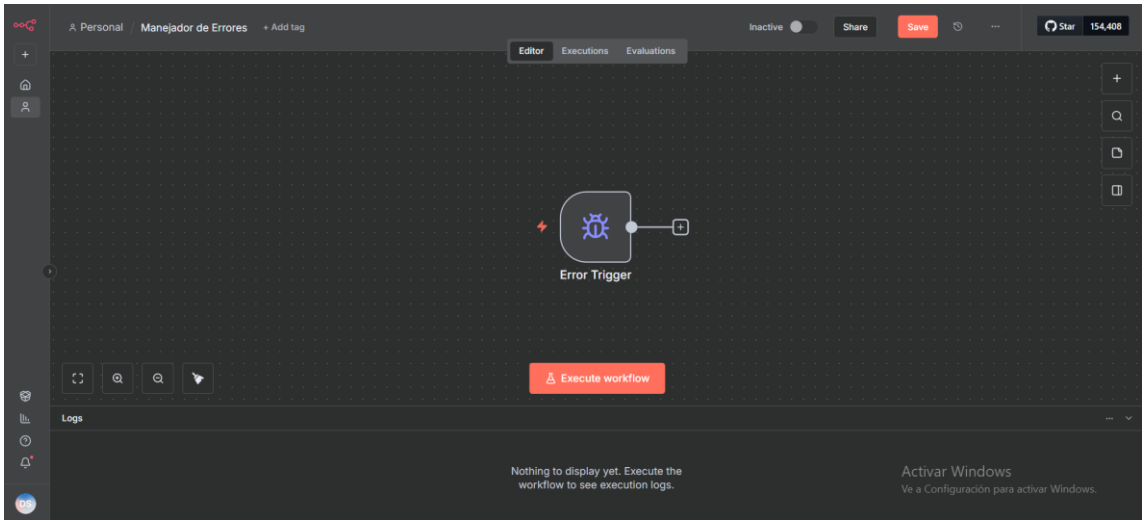
INTEGRACIÓN DE TECNOLOGÍAS Y SERVICIOS INFORMÁTICOS

Daniel Salas Alonso

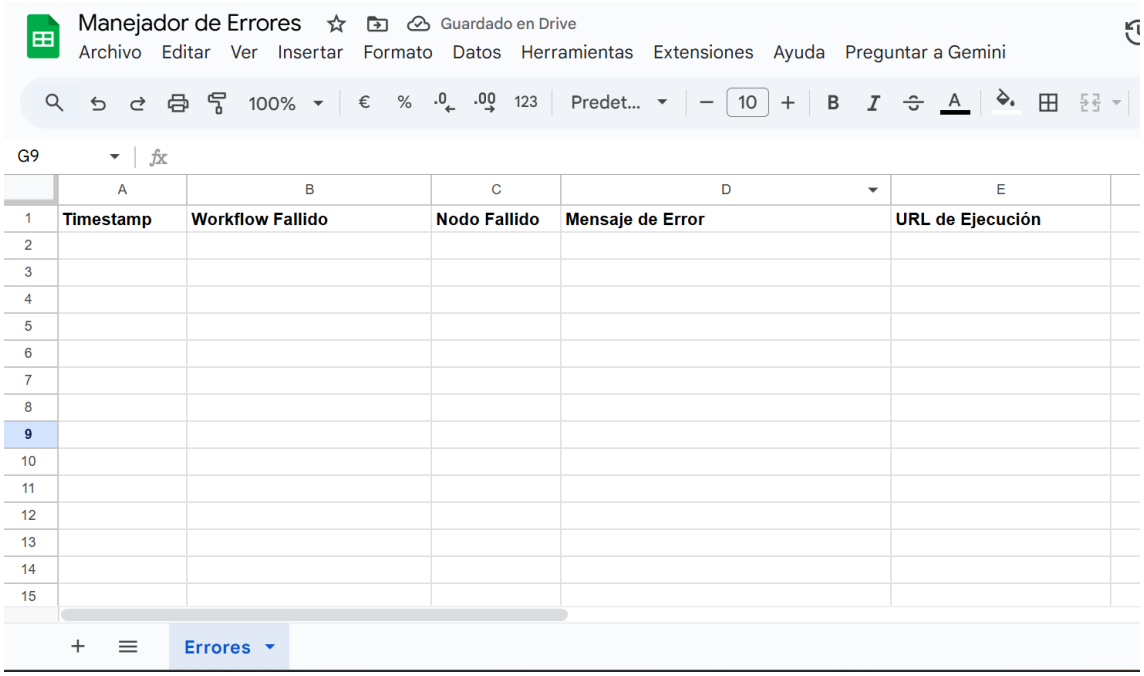
3. Desarrollo del Flujo de Trabajo Guiado

3.1. Paso 1: Crear el Flujo de Trabajo de Errores

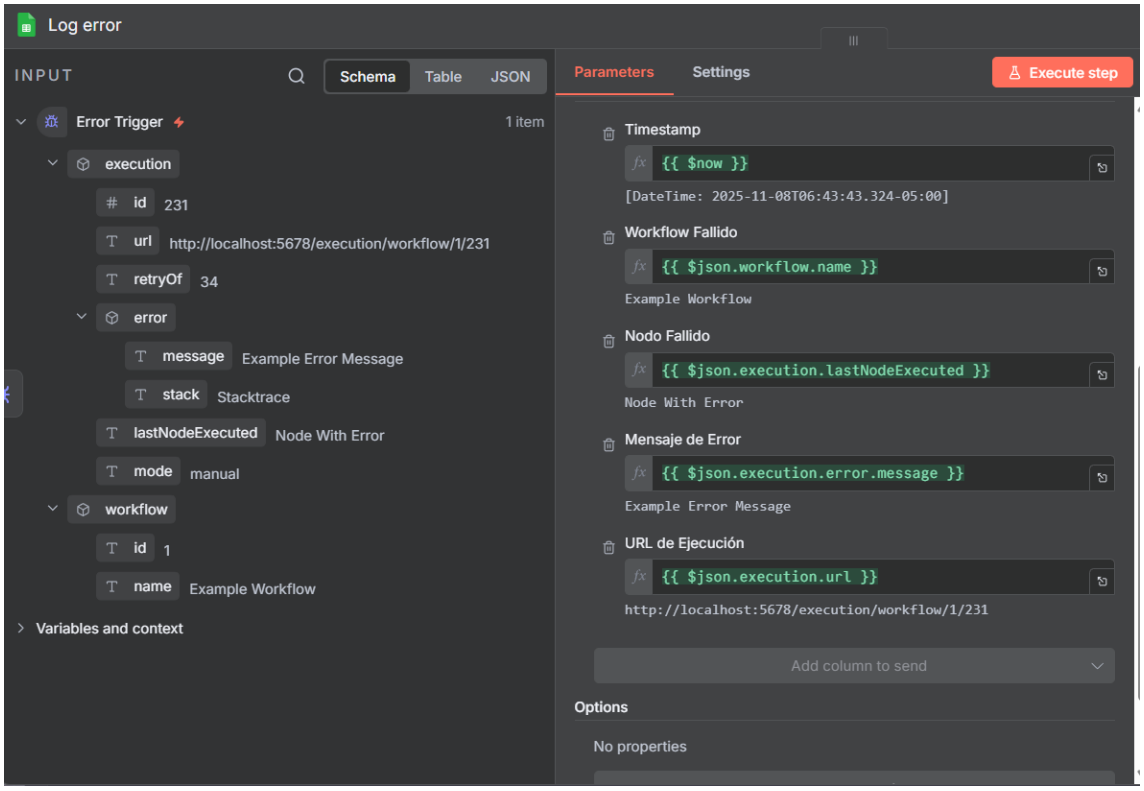
Comenzamos creando un nuevo workflow que servirá como nuestro "Manejador de Errores". El nodo inicial de este flujo es un "Error Trigger". Este nodo especial se activa automáticamente cuando otro workflow, que lo tenga configurado, falla.



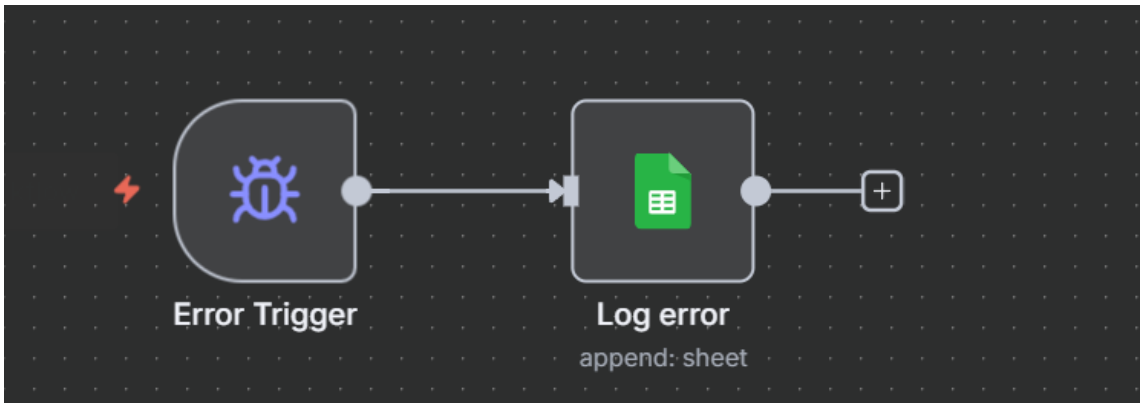
A continuación, preparamos el destino donde registraremos los fallos. Creamos un nuevo documento de Google Sheets llamado "Manejador de Errores". En una hoja (ej. "Errores"), definimos las columnas necesarias: "Timestamp", "Workflow Fallido", "Nodo Fallido", "Mensaje de Error" y "URL de Ejecución".



De vuelta en n8n, conectamos un nodo "Google Sheets" (renombrado a "Log error") a la salida del "Error Trigger". Lo configuramos en modo "Append Row" para añadir una nueva fila por cada error. Mapeamos los datos dinámicos que provee el "Error Trigger" a las columnas de nuestra hoja de cálculo. Los valores de las columnas se obtendrán de las expresiones propias del nodo anterior para "Workflow Fallido", "Nodo Fallido", "Mensaje de Error" y "URL de Ejecución", mientras que para "Timestamp" Usamos la variable global {{ $now }}.

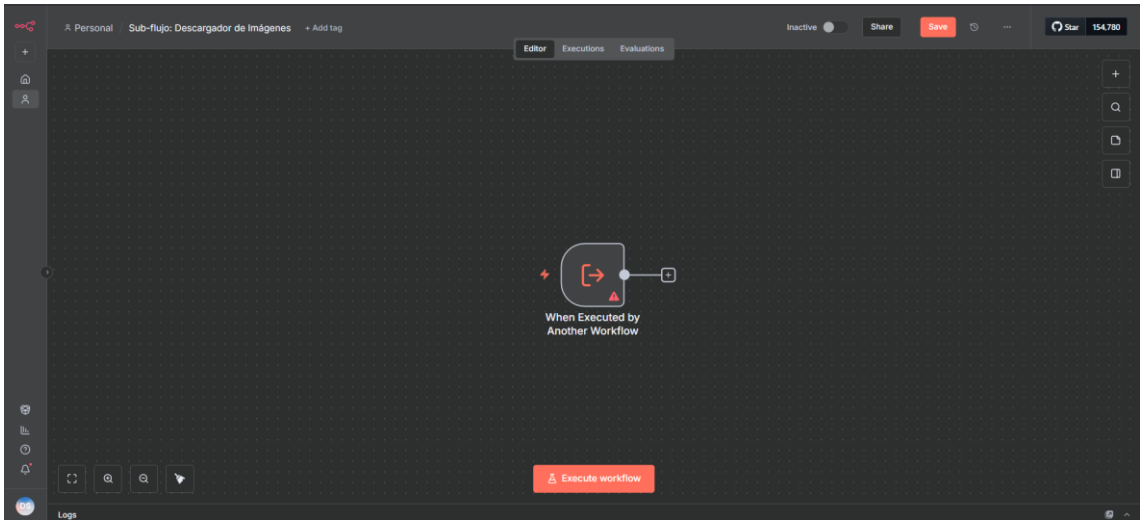


El flujo de trabajo del manejador de errores queda completo. Consiste en el "Error Trigger" que captura el fallo y el nodo "Log error" (Google Sheets) que lo registra en nuestra hoja de cálculo.

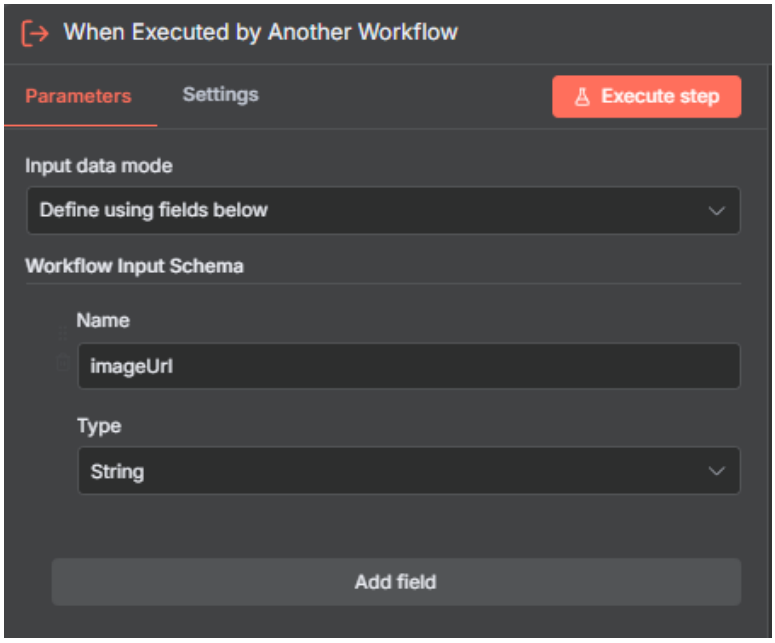


3.2. Paso 2: Crear el Sub-Flujo de Trabajo "Descargador de Imágenes"

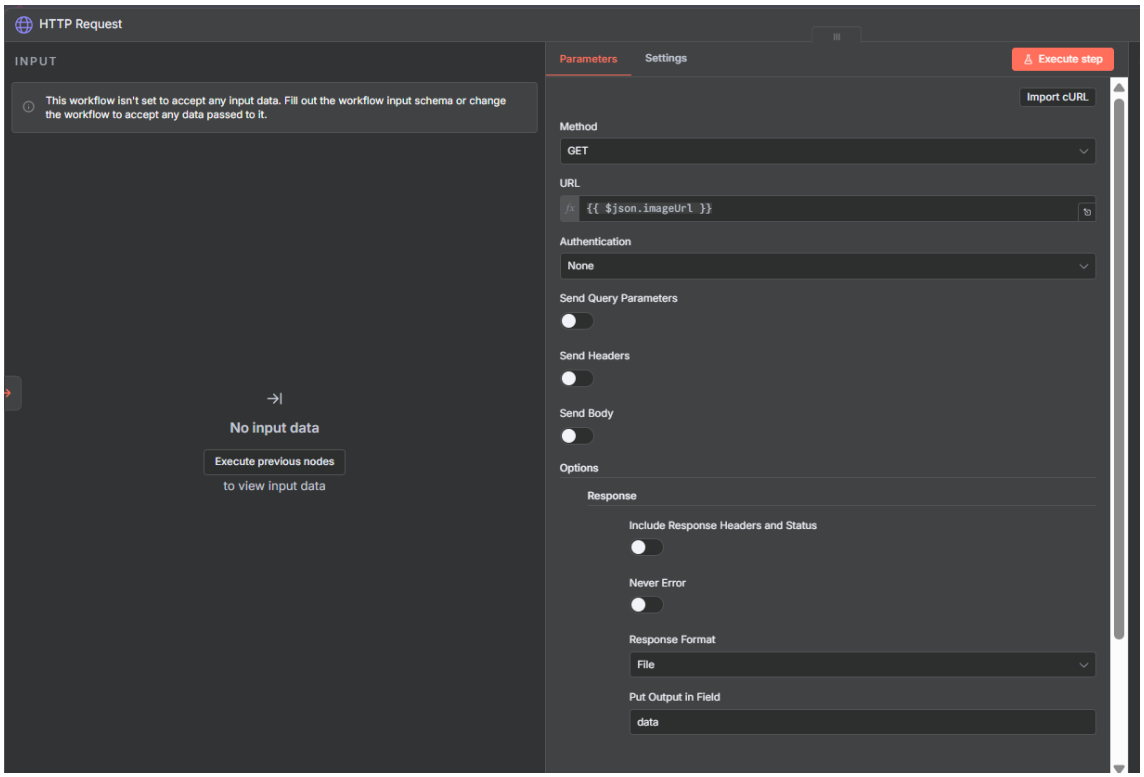
Creamos un segundo workflow que funcionará como un "sub-flujo". Lo nombramos "Sub-flujo: Descargador de Imágenes". El nodo inicial es "When Executed by Another Workflow", lo que permite que sea invocado desde un flujo de trabajo principal.



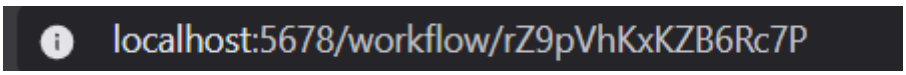
En la configuración del nodo de inicio, definimos el "Workflow Input Schema". Añadimos un campo de entrada (Field) llamado imageUrl de tipo String. Esto permite que el flujo principal pase una URL de imagen a este sub-flujo cada vez que lo llame.



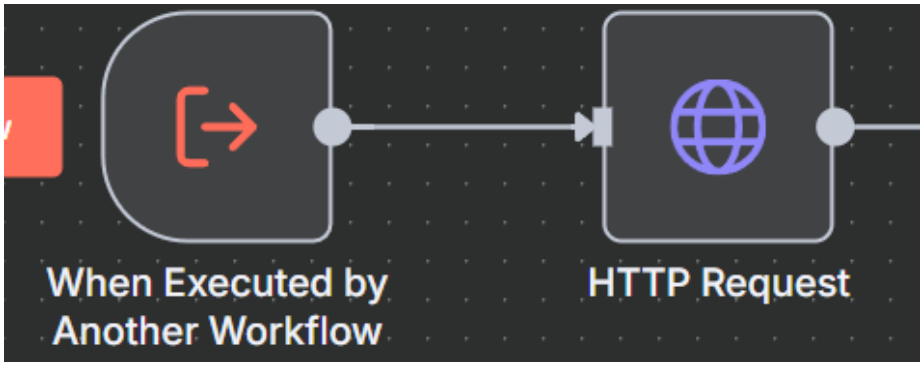
A continuación del nodo de inicio, conectamos un nodo "HTTP Request". En el campo "URL", usamos la expresión {{ $json.imageUrl }} para utilizar la URL que se recibió como entrada. Es fundamental configurar el "Response Format" como "File" para que n8n descargue la imagen como datos binarios.



Guardamos y activamos el sub-flujo. El sistema le asigna una URL única (en este caso, .../workflow/rZ9pV...). Necesitaremos esta referencia (o el nombre del workflow) para llamarlo desde el flujo principal.

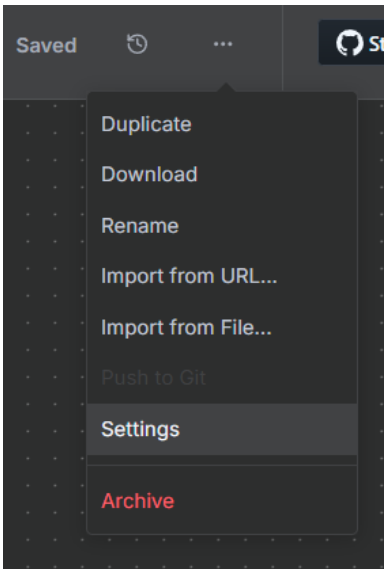


El sub-flujo "Descargador de Imágenes" queda completo. Su función es recibir una URL, descargar el archivo correspondiente mediante "HTTP Request" y devolverlo.

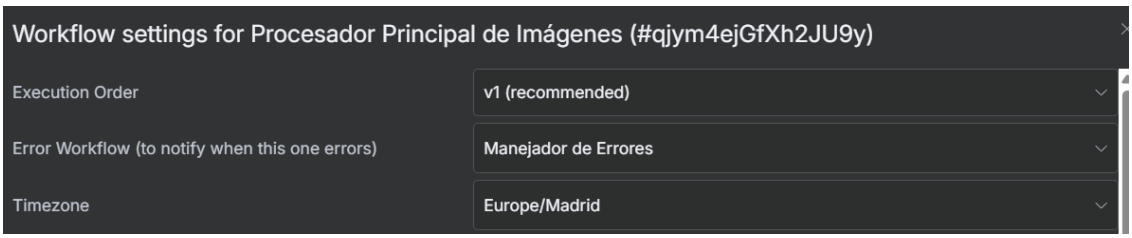


3.3. Paso 3: Crear el Flujo de Trabajo Principal

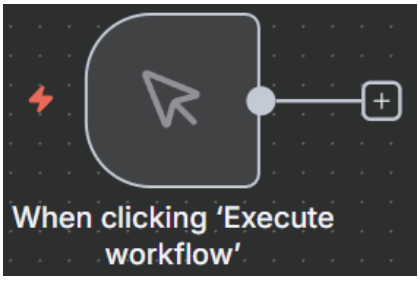
Ahora creamos un flujo de trabajo principal ("Procesador Principal de Imágenes"). Tanto en este flujo principal como en el sub-flujo que creamos antes, abrimos el menú de opciones y seleccionamos "Settings".



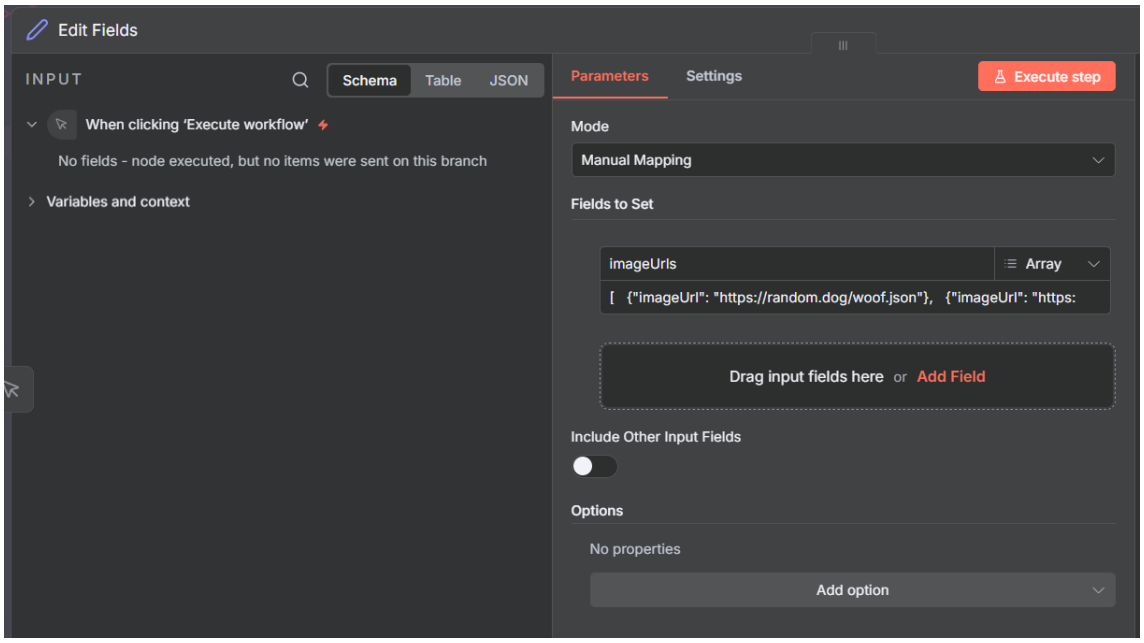
En la configuración del workflow ("Workflow settings"), buscamos la opción "Error Workflow". En el desplegable, seleccionamos el flujo "Manejador de Errores" que creamos en el Paso 1. Repetimos esta acción para ambos flujos (el principal y el sub-flujo). Esto asegura que, si cualquiera de ellos falla, se activará nuestro manejador.



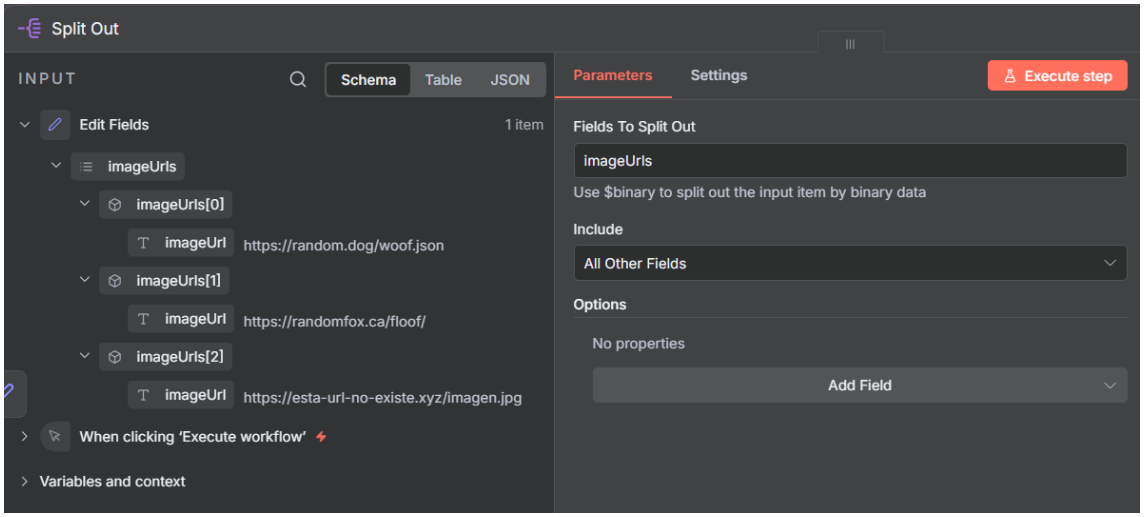
El flujo principal comienza con un "Manual Trigger", lo que nos permite iniciarlo manualmente.



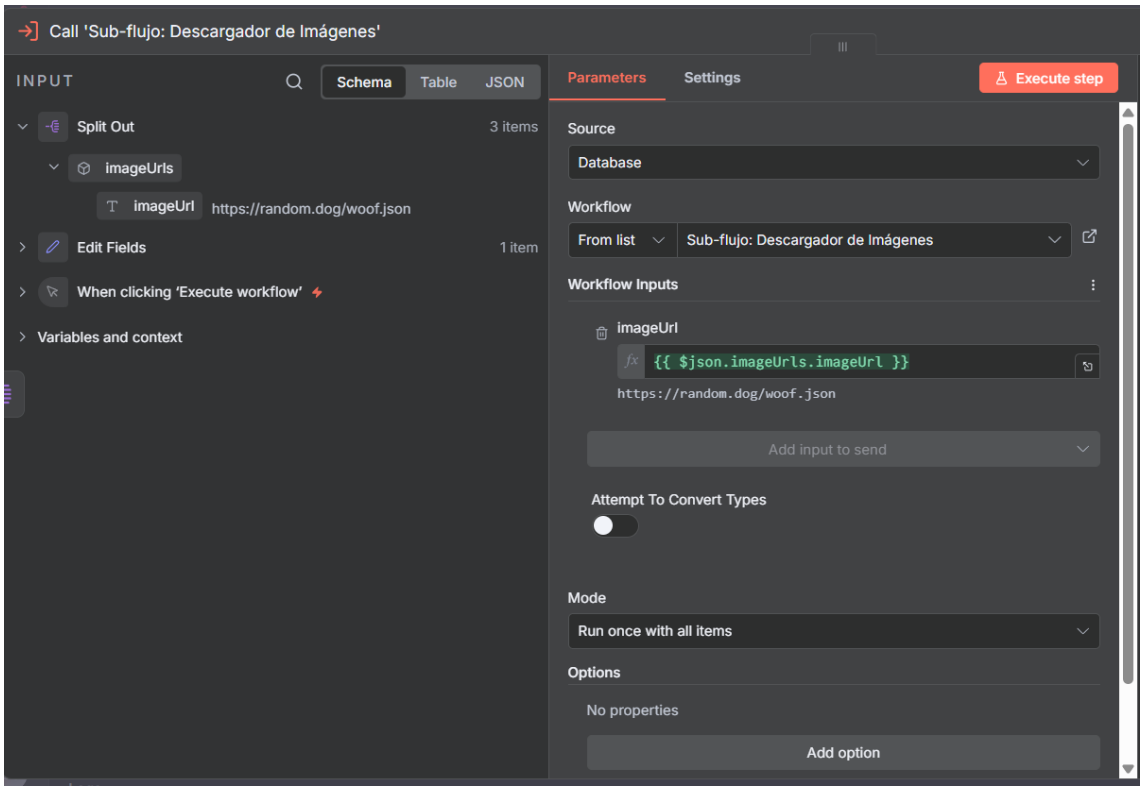
Añadimos un nodo "Edit Fields (Set)" para simular una lista de datos de entrada. Creamos un campo llamado imageUrls de tipo "Array". Dentro de este array, añadimos varias URLs. De forma intencionada, incluimos una URL incorrecta (ej. .../esta-url-no-existe.xyz/imagen.jpg) para forzar un error y probar nuestro manejador.



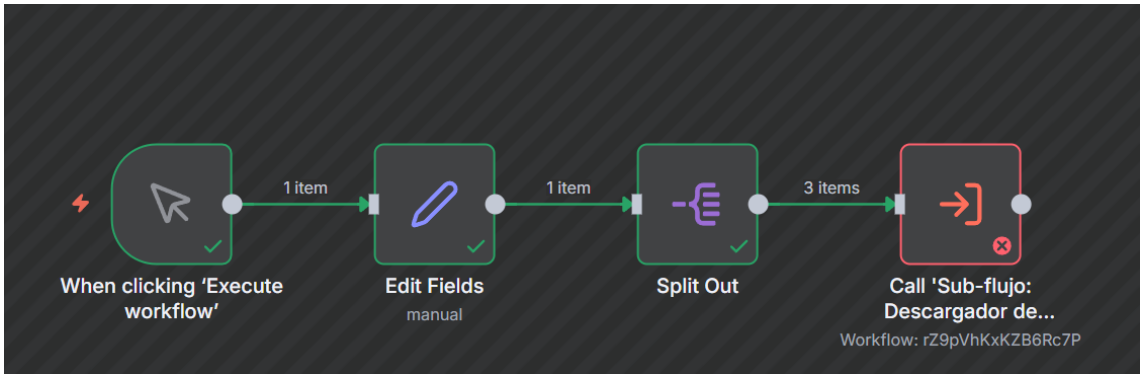
Conectamos un nodo "Split Out" a la salida del "Edit Fields". Lo configuramos para que procese el campo imageUrls. La función de este nodo es tomar el array de URLs y separarlo en ítems individuales (uno por cada URL), para procesarlos uno por uno.



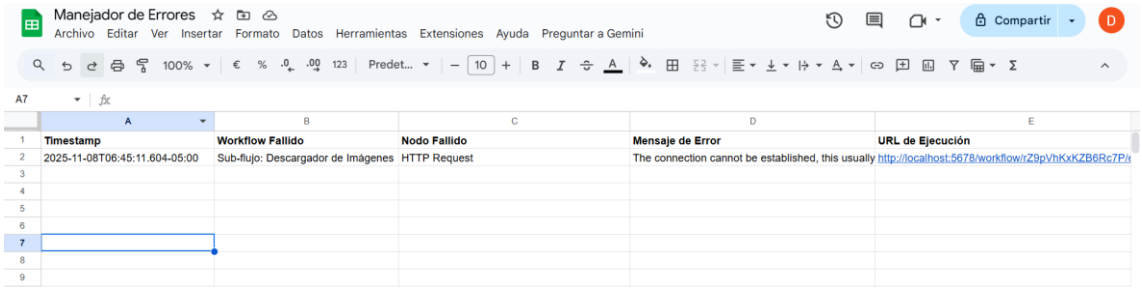
Añadimos un nodo "Execute Workflow". En la configuración, seleccionamos nuestro "Sub-flujo: Descargador de Imágenes". Luego, en "Workflow Inputs", mapeamos el campo de entrada imageUrl del sub-flujo con la salida del "Split Out".



Vemos el flujo principal completo. Al ejecutarlo, el "Split Out" pasa las tres URLs, una por una, al nodo "Call Descargador de Imágenes ". Las primeras dos ejecuciones del sub-flujo tienen éxito, pero la tercera (con la URL rota) falla. Este fallo se marca en rojo, detiene la ejecución y activa nuestro "Manejador de Errores".



Finalmente, verificamos nuestro Google Sheet "Manejador de Errores". Observamos que se ha añadido una nueva fila. La fila registra correctamente que el "Workflow Fallido" fue el "Sub-flujo: Descargador de Imágenes", el "Nodo Fallido" fue "HTTP Request" (el nodo que falló dentro del sub-flujo), y el "Mensaje de Error" es "The connection cannot be established...", confirmando que el sistema funcionó.

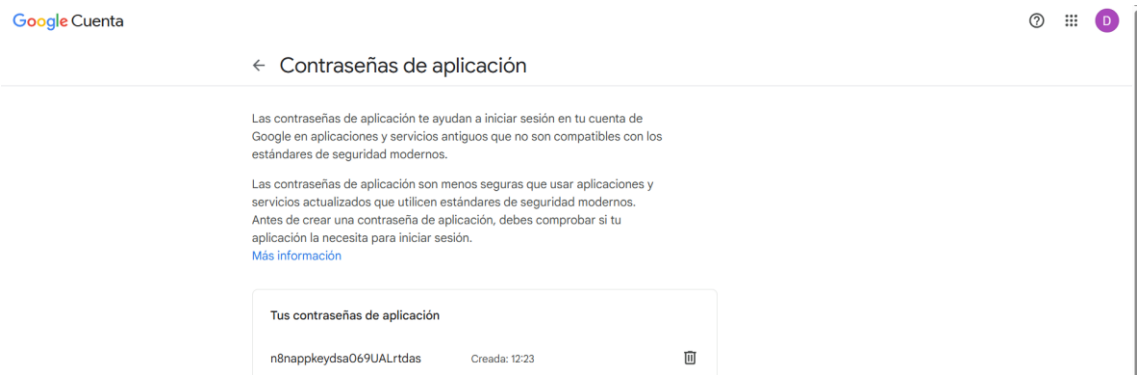


4. Ejercicios Propuestos

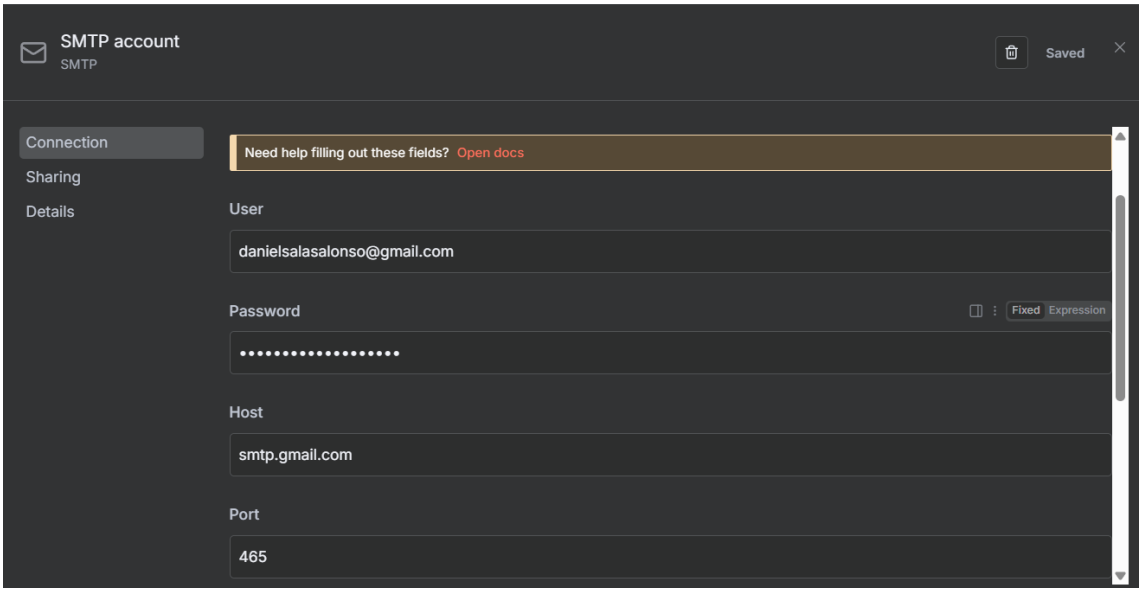
4.1. Ejercicio 1: Notificación de Errores Mejorada (Dificultad: Baja)

1. *Modifique el flujo de trabajo "Manejador de Errores" y añada un nodo Send Email (o un nodo de Slack/Telegram si lo prefiere)*

Para este ejercicio, Duplicamos el workflow "Manejador de Errores" a Manejador de Errores 4.1” para que también envíe un email. Como vamos a usar Gmail, primero debemos ir a la configuración de seguridad de nuestra cuenta de Google y generar una "Contraseña de aplicación" para n8n, ya que las credenciales SMTP estándar no funcionarán con OAuth.

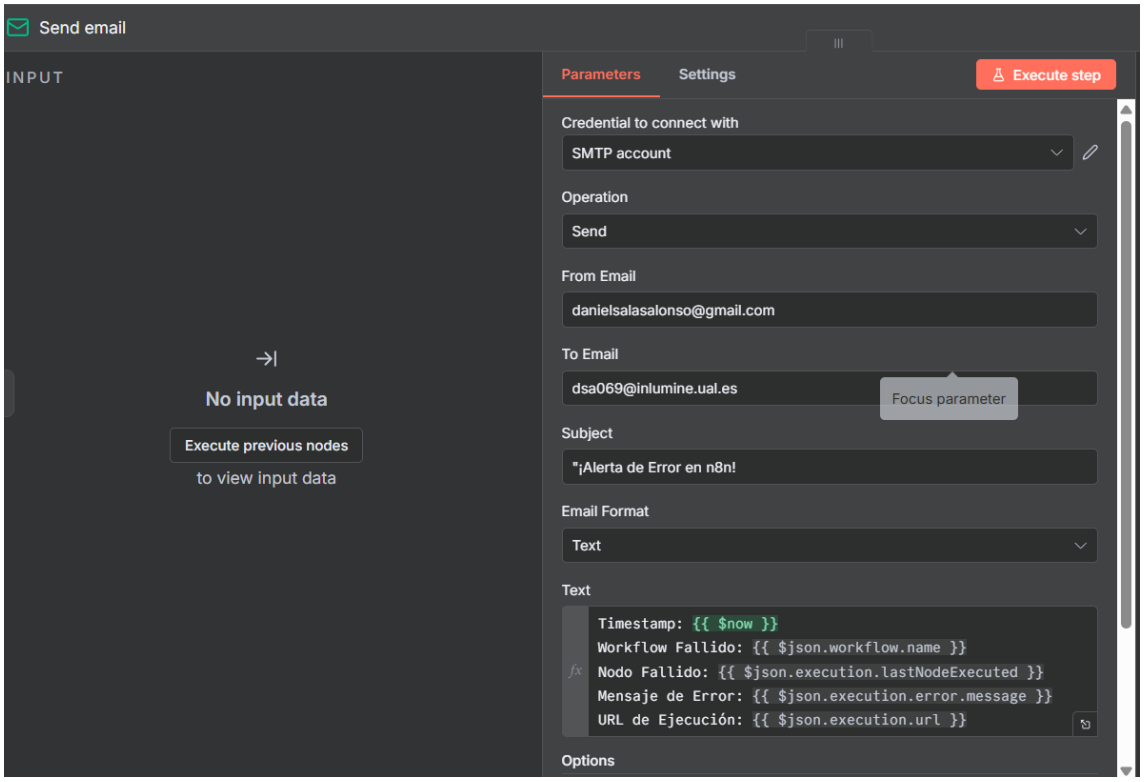


En n8n, vamos a la sección "Credentials" y creamos una nueva credencial de tipo "SMTP account". Rellenamos los campos: "User" (nuestro email, ej. danielsalasalonso@gmail.com), "Password" (la contraseña de aplicación generada en el paso anterior), "Host" (smtp.gmail.com) y "Port" (465).

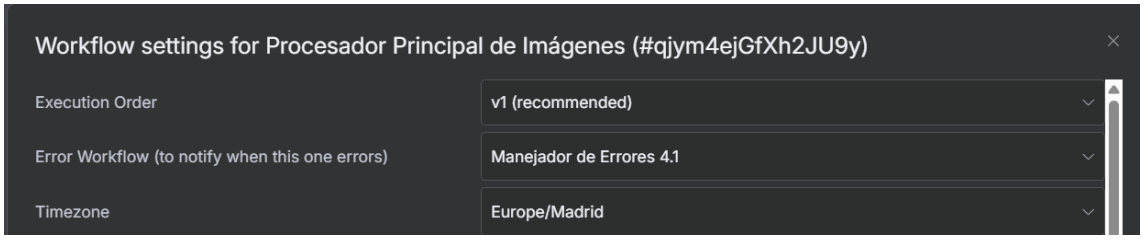


2. *Configure el nodo para enviar un correo con un asunto claro (ej. "¡Alerta de Error en n8n!"). En el cuerpo del correo, incluya la misma información que se registra en Google Sheets (nombre del flujo, nodo, mensaje y URL), utilizando las expresiones del Error Trigger.*

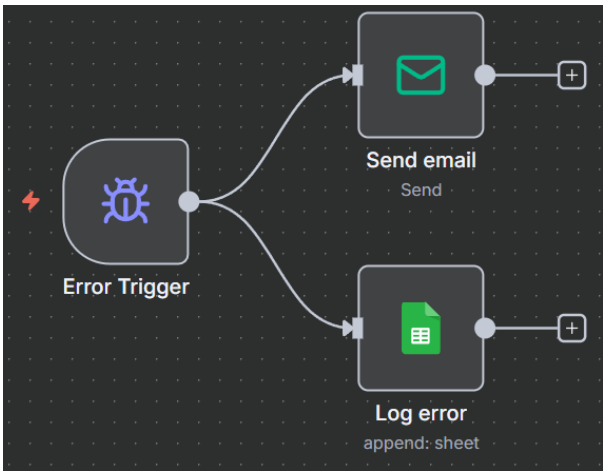
Volvemos a nuestro workflow "Manejador de Errores". Añadimos un nodo "Send email". Seleccionamos la credencial SMTP que acabamos de crear. Configuramos el "To Email" (ej. dsa069@inlumine.ual.es), el "Subject" (ej. "¡Alerta de Error en n8n!") y el "Text" (cuerpo del correo). En el cuerpo, usamos las mismas expresiones del "Error Trigger", utilizadas para el nodo Google Sheets, para incluir todos los detalles del fallo.



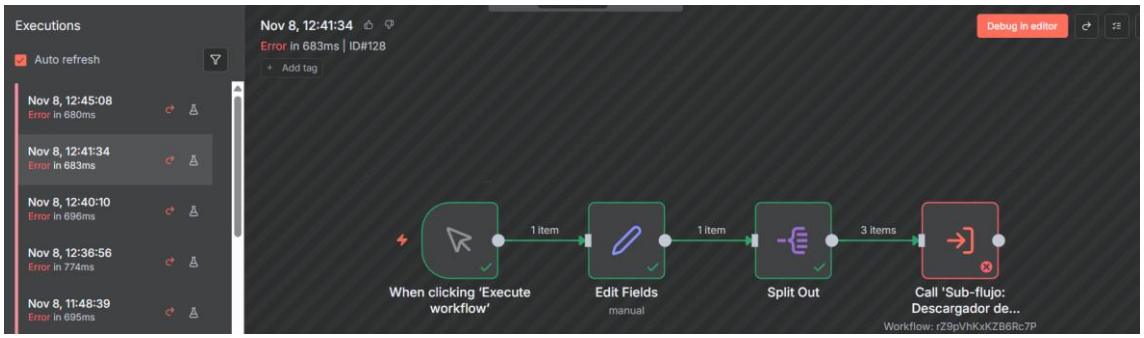
Guardamos este manejador mejorado (ej. "Manejador de Errores 4.1") y nos aseguramos de que nuestro flujo principal y sus Sub-Flujos (en "Settings") estén apuntando a esta nueva versión del manejador de errores.



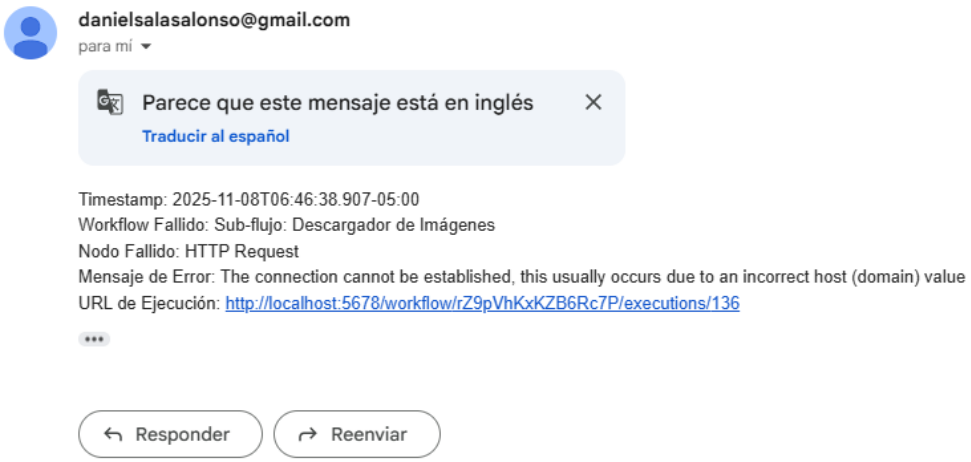
El "Error Trigger" ahora tiene dos ramas de salida. La ejecución se bifurca y los datos del error se envían simultáneamente al nodo "Send email" (para notificación inmediata) y al nodo "Log error" (para el registro persistente en Google Sheets).



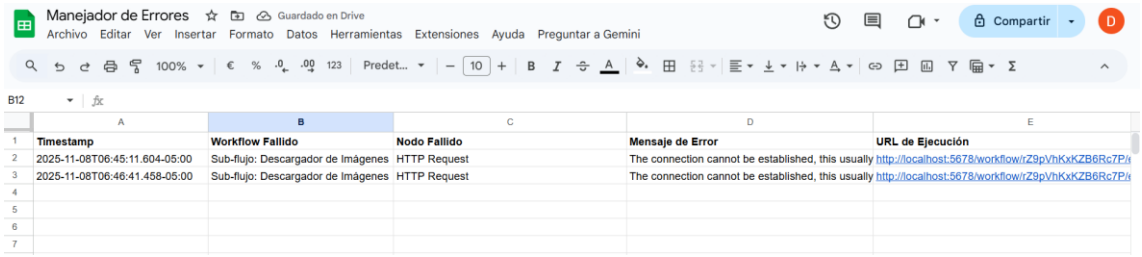
Volvemos a ejecutar el flujo principal con la URL inexistente.



Revisamos la bandeja de entrada del correo de destino. Verificamos que hemos recibido la alerta por correo electrónico. El cuerpo del mensaje contiene todos los detalles del error que configuramos (Timestamp, Workflow Fallido, Nodo Fallido, Mensaje de Error y URL de Ejecución), confirmando que la notificación funciona.



Comprobamos también el Google Sheets. Vemos que el error de esta nueva ejecución también ha sido registrado (ahora hay dos entradas idénticas de las dos pruebas). El sistema funciona en paralelo, notificando y registrando.



Manejador de Errores

Guardado en Drive

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda Preguntar a Gemini

Compartir

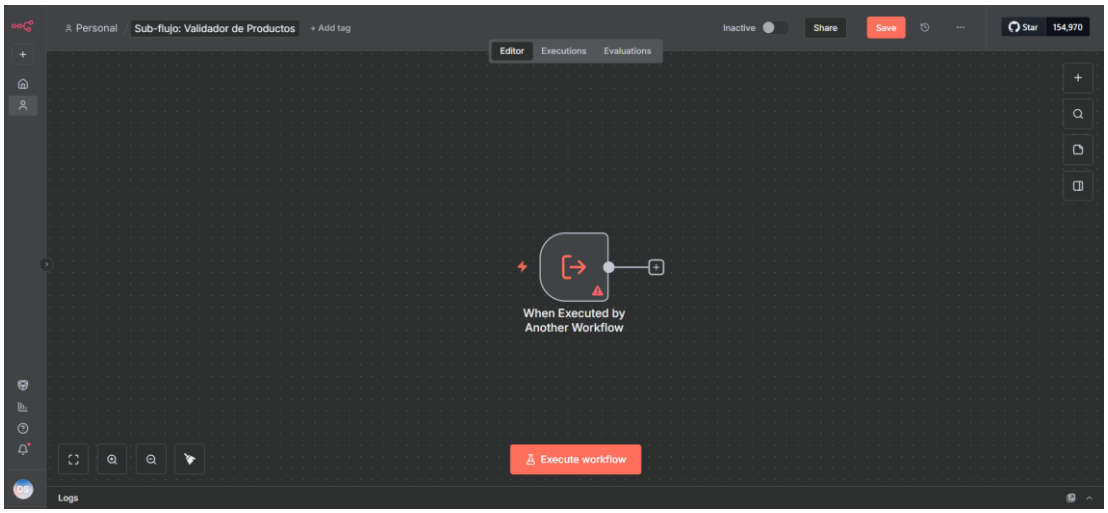
B12

	A	B	C	D	E
1	**Timestamp**	**Workflow Fallido**	**Nodo Fallido**	**Mensaje de Error**	**URL de Ejecución**
2	2025-11-08T06:45:11.604-05:00	Sub-flujo: Descargador de Imágenes	HTTP Request	The connection cannot be established, this usually	http://localhost:5678/workflow/rZ9pVhKxKZB6Rc7P/e
3	2025-11-08T06:46:41.458-05:00	Sub-flujo: Descargador de Imágenes	HTTP Request	The connection cannot be established, this usually	http://localhost:5678/workflow/rZ9pVhKxKZB6Rc7P/e
4					
5					
6					
7					

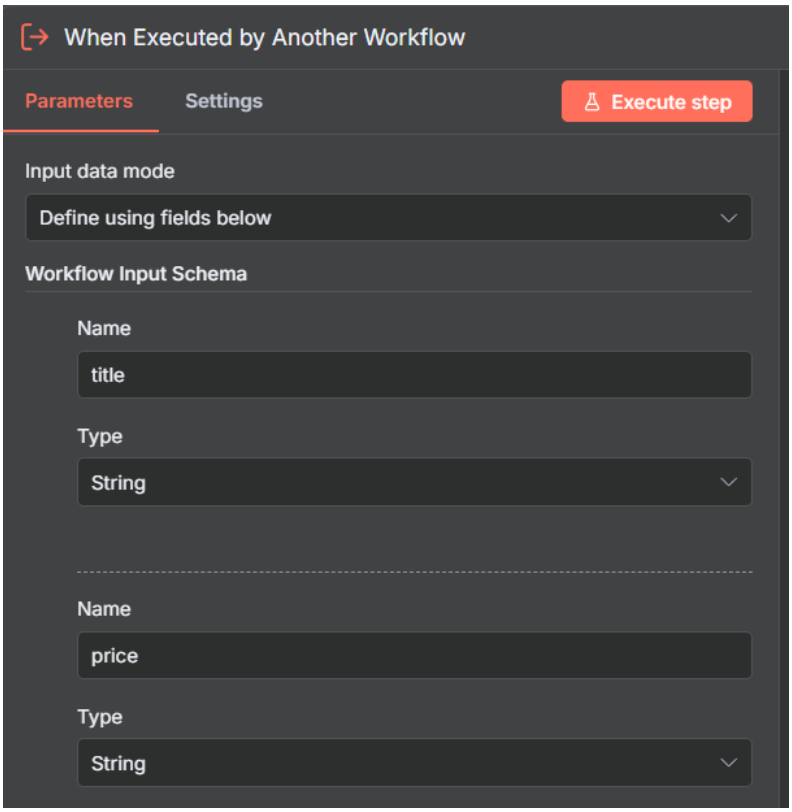
4.2. Ejercicio 2: Sub-Flujo de Validación de Datos (Dificultad: Media)

1. *Cree un sub-flujo llamado "Sub-flujo: Validador de Productos" que comience con un Execute Workflow Trigger.*

Creamos un nuevo sub-flujo llamado "Sub-flujo: Validador de Productos". Su nodo inicial es "When Executed by Another Workflow".

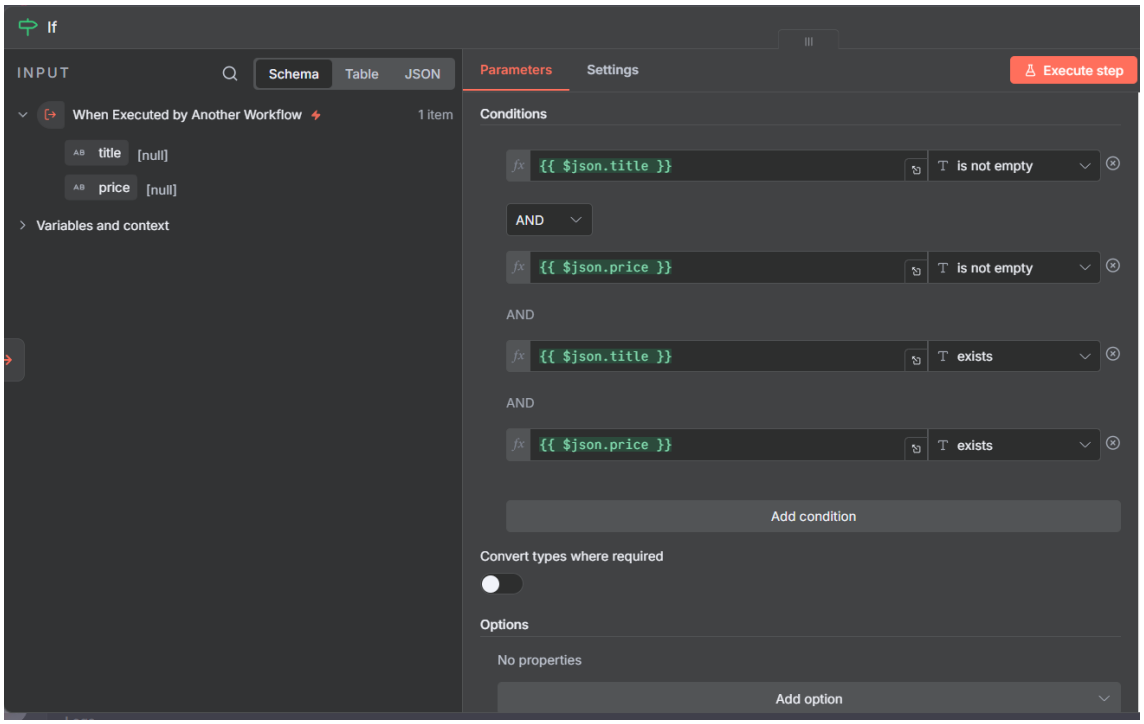


Configuramos el "Workflow Input Schema" del nodo inicial para que acepte dos campos de entrada que queremos validar: title (String) y price (String).



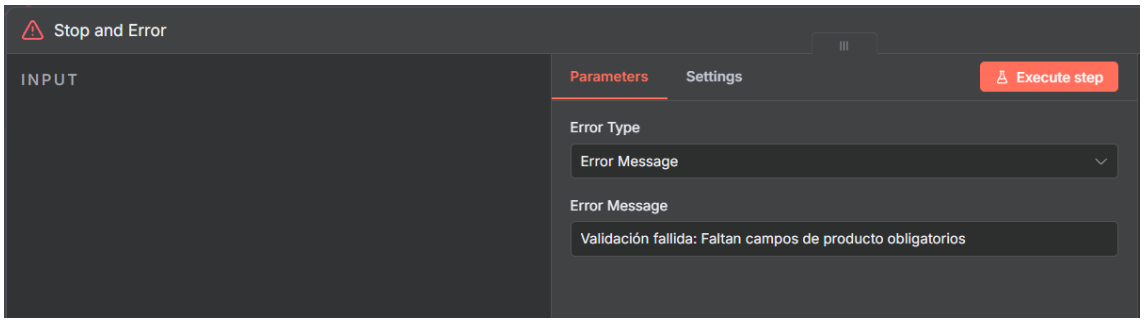
2. *Dentro de este sub-flujo, use un nodo IF para comprobar si los campos title Y price existen y no están vacíos. (Pista: puede necesitar varias condiciones con el operador AND).*

Añadimos un nodo "IF" para realizar la validación. Configuramos múltiples condiciones con el operador "AND" y comprobamos que el precio y el titulo sean campos existentes y que no estén vacíos. Solo si se cumplen las cuatro condiciones, la ejecución saldrá por la rama "true".

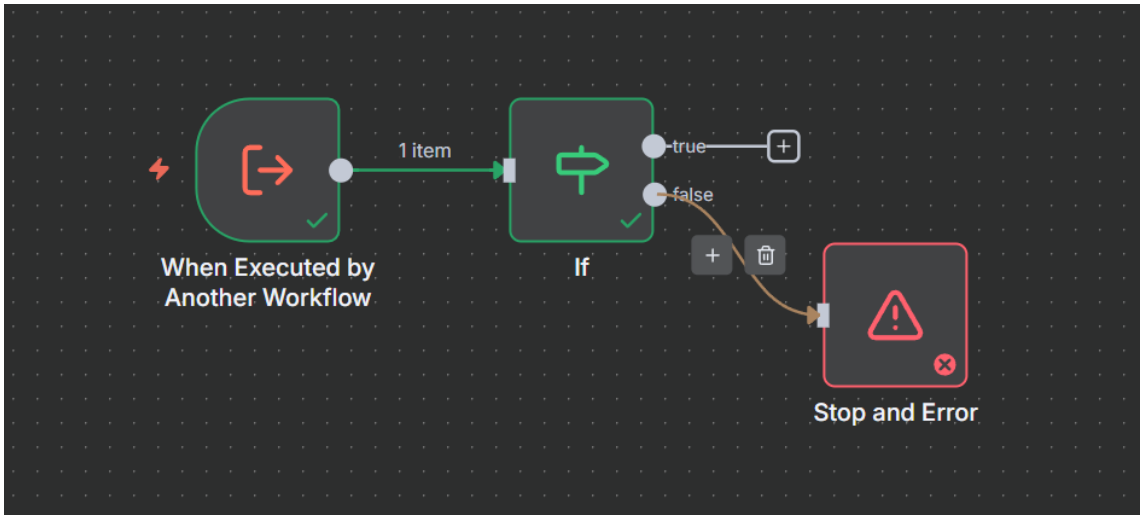


3. *Si la condición es false (falta alguno de los campos), conecte un nodo Stop and Error con el mensaje "Validación fallida: Faltan campos de producto obligatorios".*

Conectamos un nodo "Stop and Error" a la salida "false" del nodo IF. Si title o price faltan, la ejecución irá por esta rama. Establecemos un "Error Message" personalizado: "Validación fallida: Faltan campos de producto obligatorios". Este nodo detendrá el sub-flujo y generará un error con este mensaje específico.

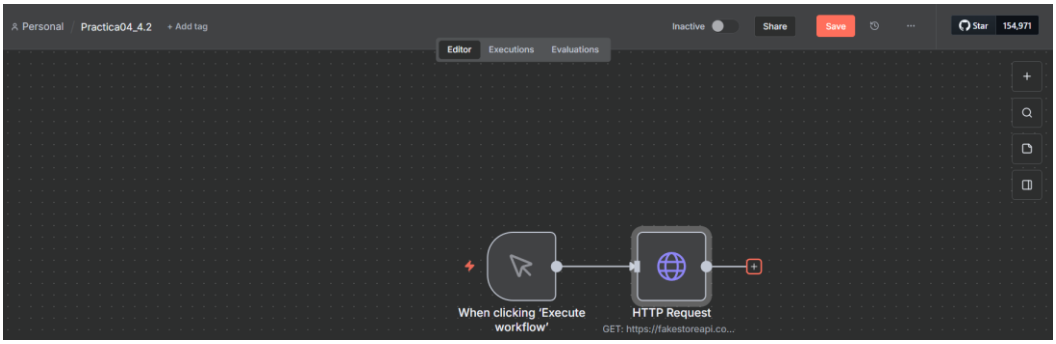


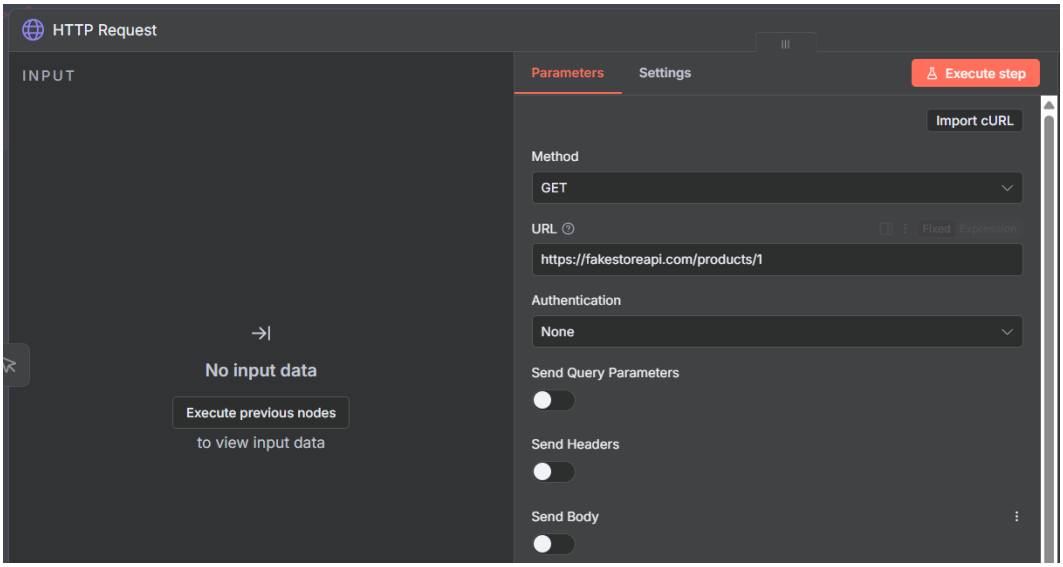
El sub-flujo validador queda completo. Si la validación (IF) es exitosa, el flujo termina (rama "true"). Si la validación falla, genera un error (rama "false").



4. *Cree un flujo principal que llame a la Fake Store API para obtener un producto.*

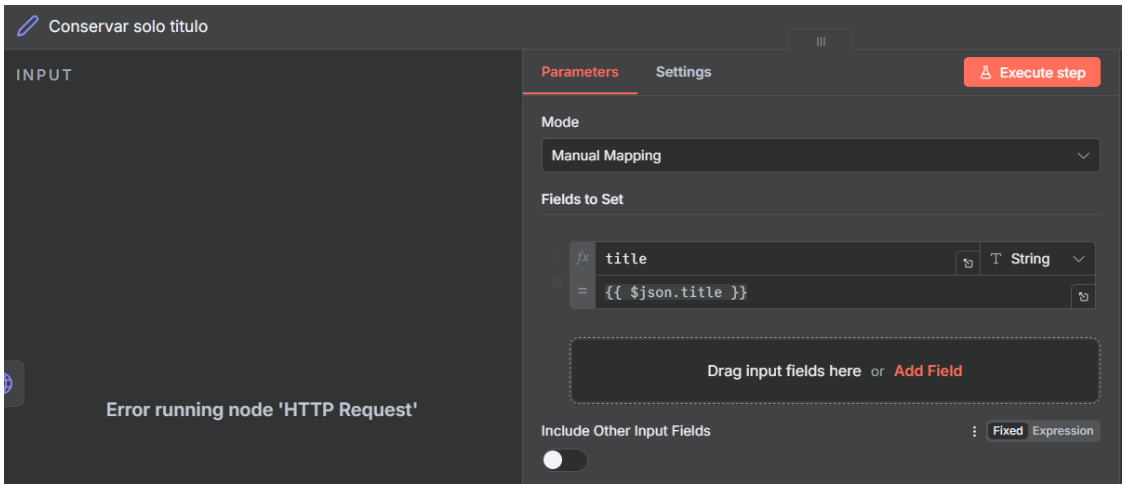
Creamos un nuevo flujo principal ("Practica04_4.2"). Comienza con un "Manual Trigger" y un "HTTP Request" que obtiene un solo producto de la Fake Store API.





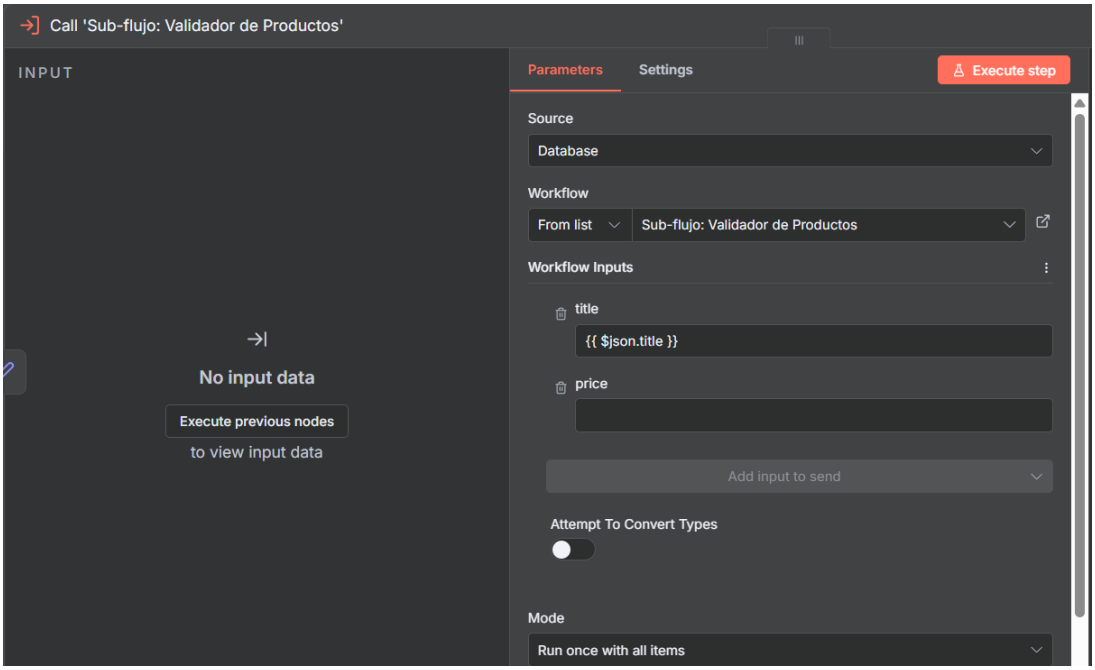
5. *(Para forzar el error) Añada un nodo Edit Fields (Set) después de la llamada a la API para eliminar el campo price del objeto del producto.*

Para forzar el error de validación, añadimos un nodo "Edit Fields (Set)". En este mapeamos únicamente el campo title de la salida anterior. No marcamos "Include Other Input Fields". Esto significa que solo el campo title pasará al siguiente nodo; el campo price se descarta.



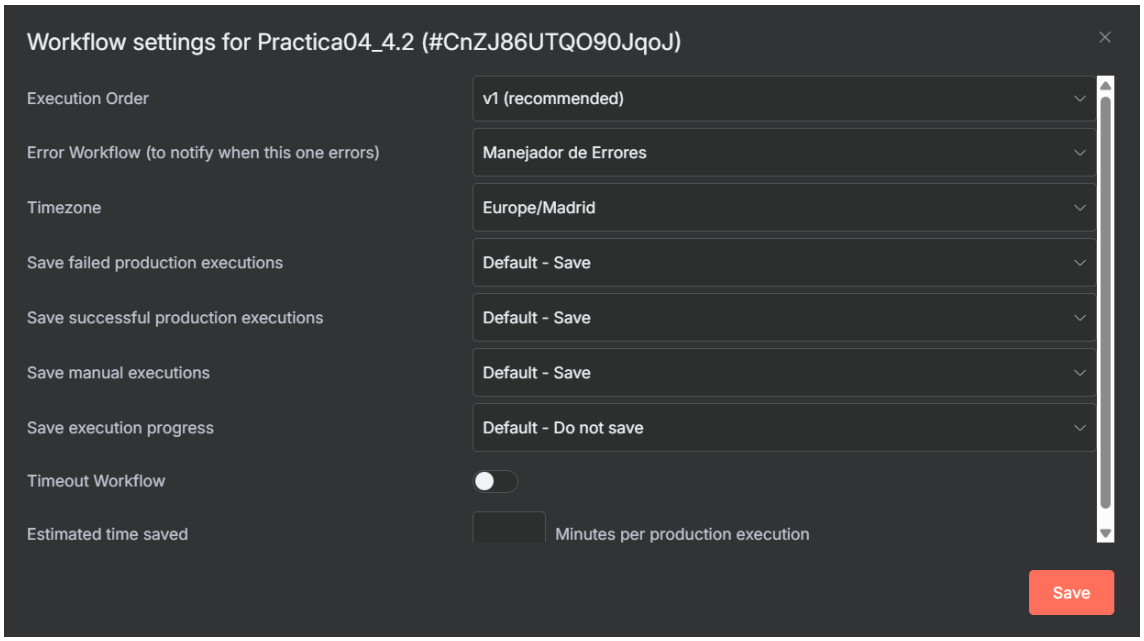
6. *Llame al sub-flujo "Validador de Productos" con el nodo Execute Workflow.*

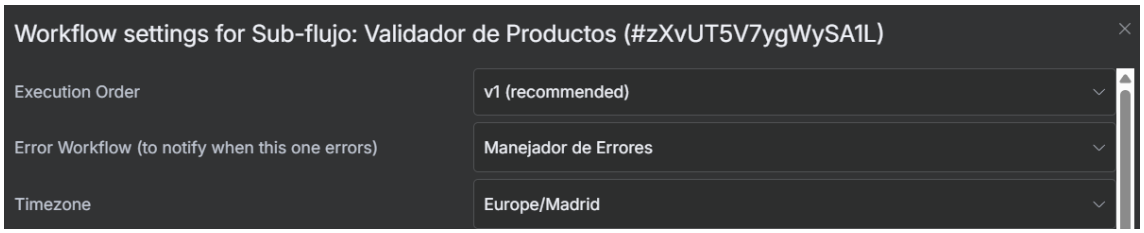
Añadimos un nodo "Execute Workflow" para llamar a nuestro sub-flujo validador. Al llamar a nuestro sub-flujo, únicamente le pasamos por entrada el campo title, el cual es el único campo que recibimos del nodo anterior, dejando la entrada de price vacía. El sub-flujo recibe title {{ $json.title }} (que existe) y price recibe {{ $json.price }} (que ahora es nulo o inexistente).



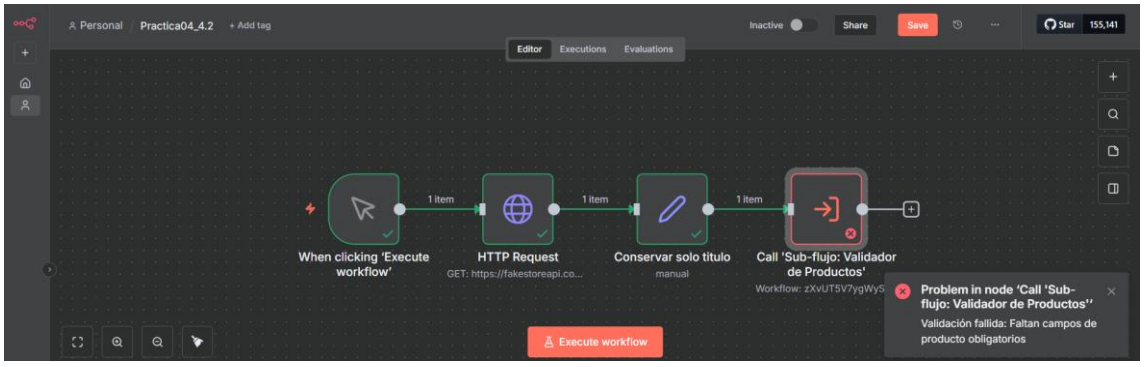
7. *Asegúrese de que el flujo principal tiene configurado el "Manejador de Errores" y verifique que el error personalizado se registra correctamente en su hoja de Google Sheets.*

Nos aseguramos de que tanto el flujo principal como el "Sub-flujo: Validador de Productos" tengan su "Error Workflow" configurado para usar nuestro "Manejador de Errores"

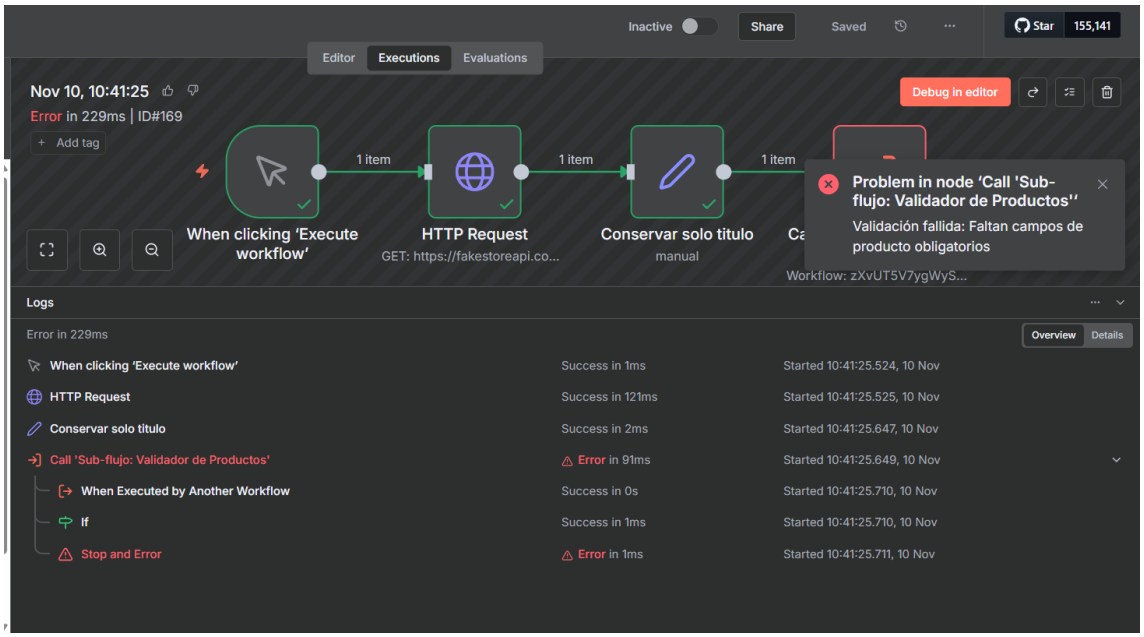




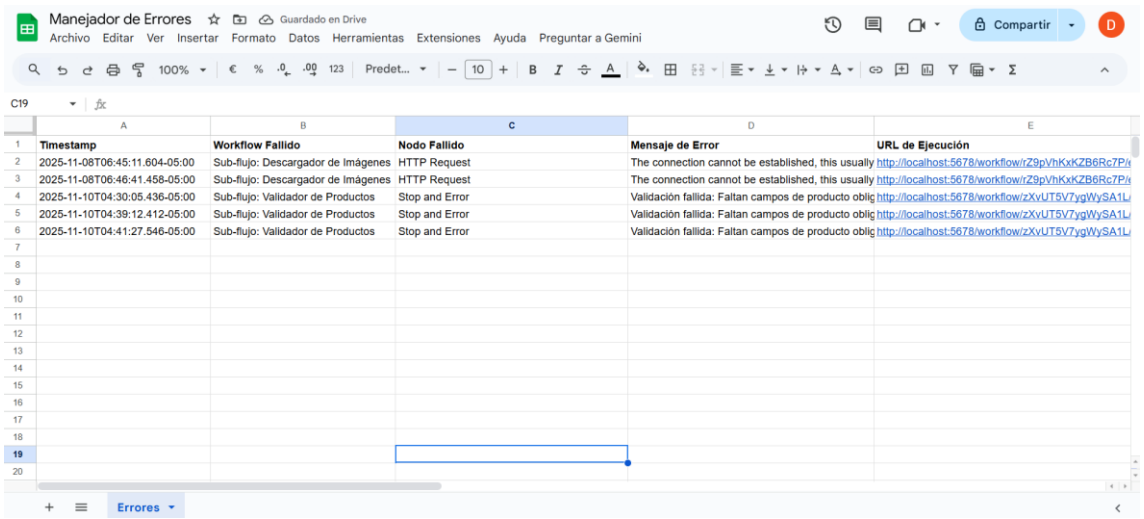
Vemos el flujo principal completo. Al ejecutarlo, el nodo "Call 'Sub-flujo: Validador de Productos'" falla. La burbuja de error muestra nuestro mensaje personalizado: "Validación fallida: Faltan campos de producto obligatorios", que se originó en el nodo "Stop and Error" del sub-flujo.



Al inspeccionar los detalles de la ejecución, vemos la traza: el flujo principal (arriba) llama al "Call 'Sub-flujo...'". Dentro del sub-flujo (abajo), el nodo "If" evalúa como "false" (porque price no existe) y la ejecución se desvía al "Stop and Error", que es el nodo que falla.



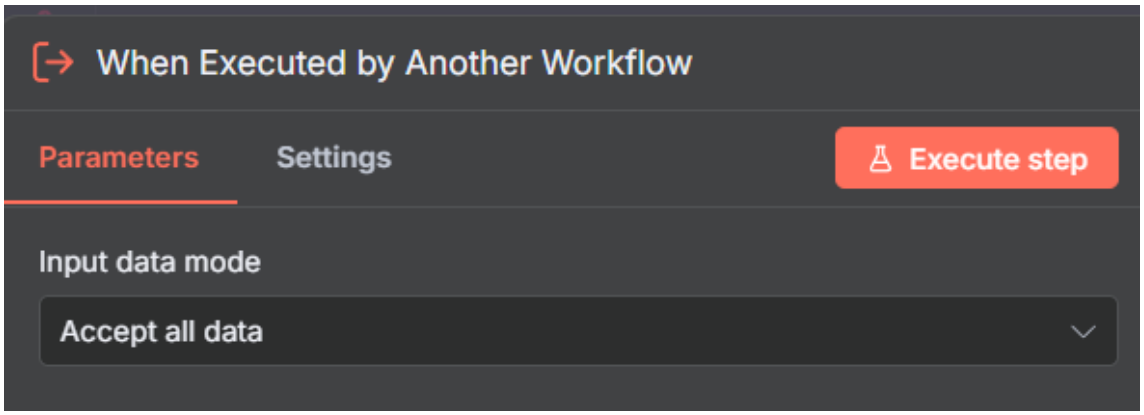
Revisamos el Google Sheets "Manejador de Errores". Vemos las nuevas entradas: el "Workflow Fallido" es "Sub-flujo: Validador de Productos", el "Nodo Fallido" es "Stop and Error", y el "Mensaje de Error" es nuestro mensaje personalizado. Esto confirma que podemos crear y capturar errores de validación personalizados.



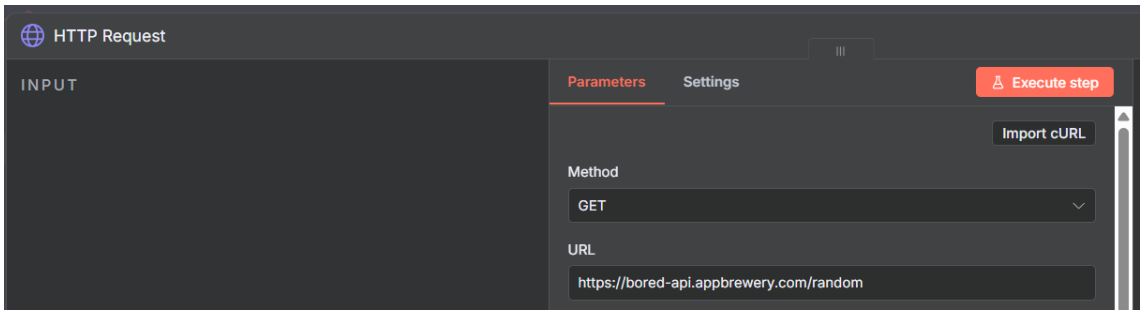
4.3. Ejercicio 3: Orquestador de Tareas Dinámicas (Dificultad: Alta)

1. *Cree dos sub-flujos simples:*
    - *"Sub-flujo: Obtener Actividad": Llama a la Bored API y devuelve la actividad.*

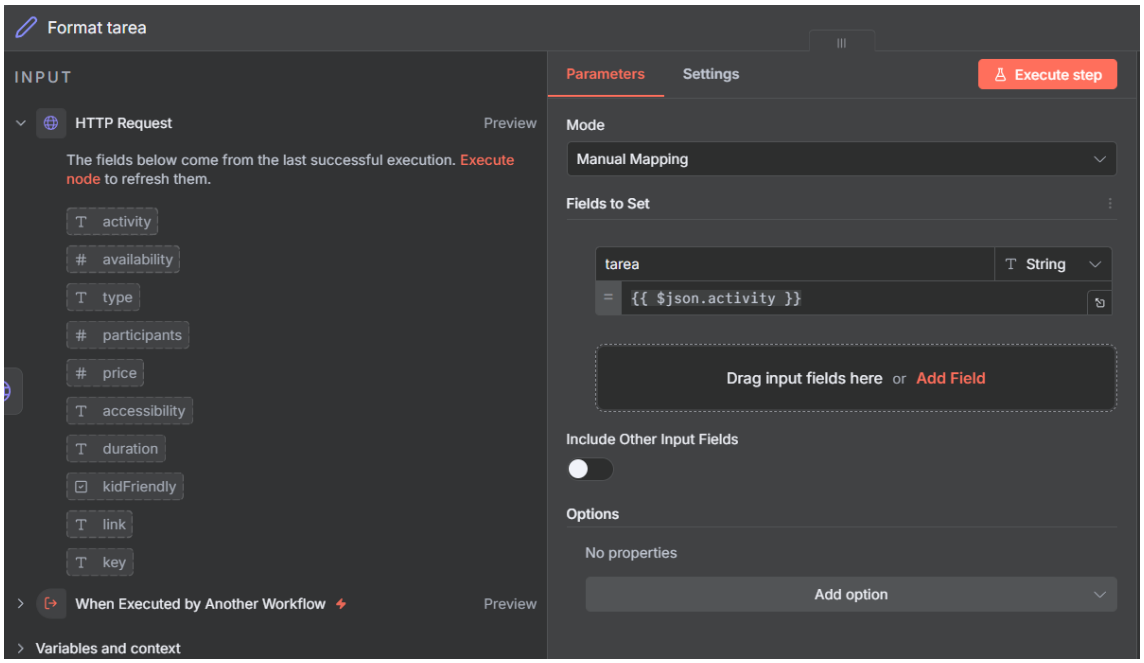
Creamos el primer sub-flujo, "Sub-flujo: Obtener Actividad". Añadimos el nodo inicial "When Executed by Another Workflow", configurado en modo "Accept all data".



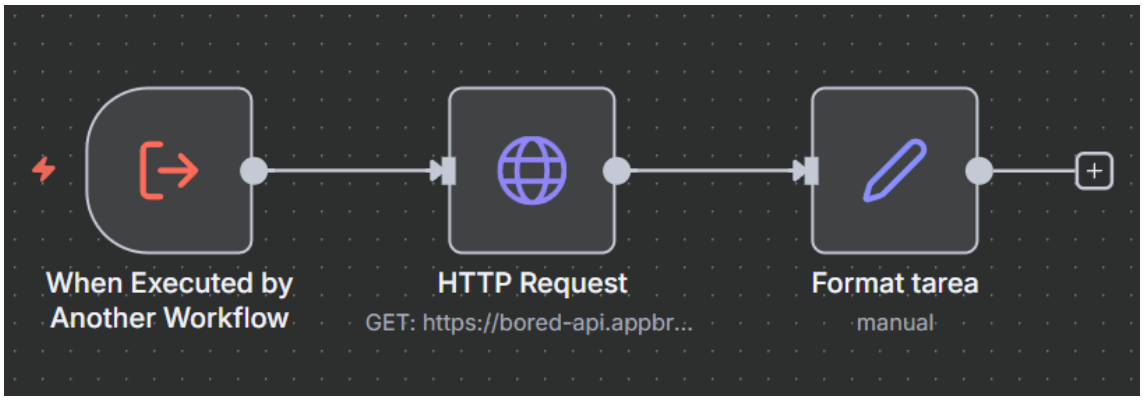
A continuación, añadimos el nodo HTTP Request, el cual, llama a la Bored API para obtener una actividad aleatoria.



Creamos un nodo Edit Fields para que el resultado siempre se devuelva bajo la variable tarea, al flujo principal, independientemente del sub-flujo, creamos un campo tarea y le asignamos el valor del campo actividad obtenido en el nodo anterior.

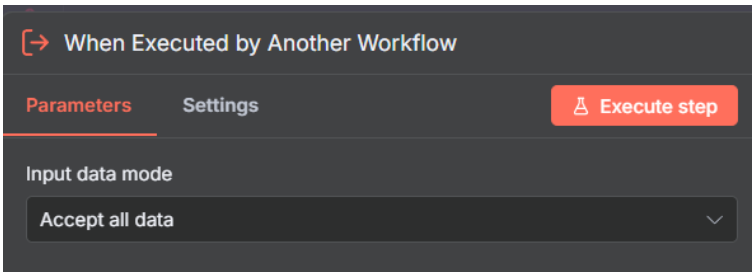


Observamos la estructura final de nuestro workflow:

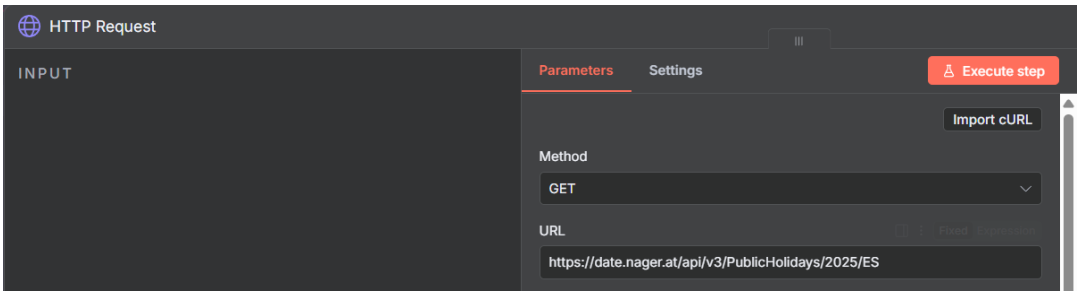


- *"Sub-flujo: Obtener Festivos": Llama a la API de festivos y devuelve la lista de festivos.*

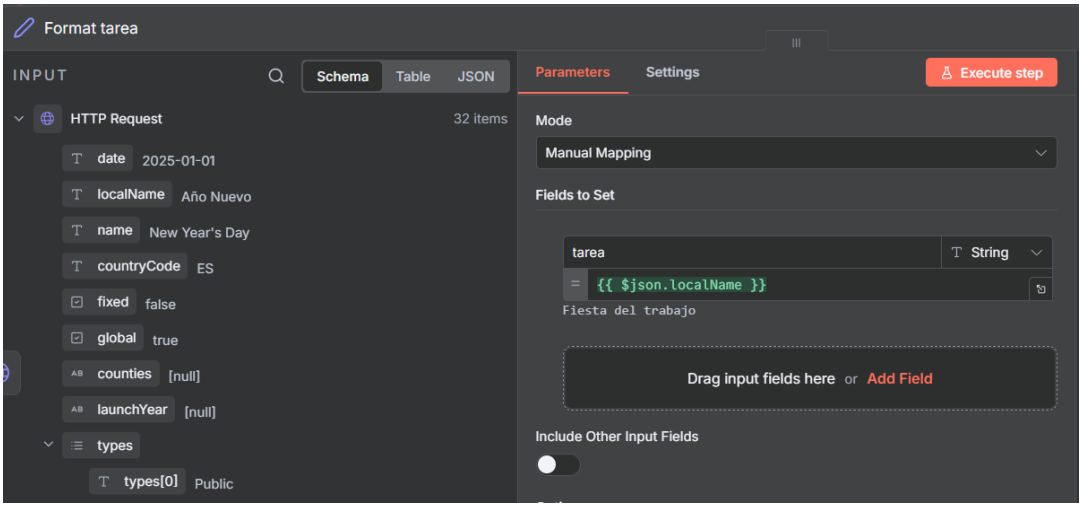
Creamos el primer sub-flujo, "Sub-flujo: Obtener Festivos". Añadimos el nodo inicial "When Executed by Another Workflow", configurado en modo "Accept all data".



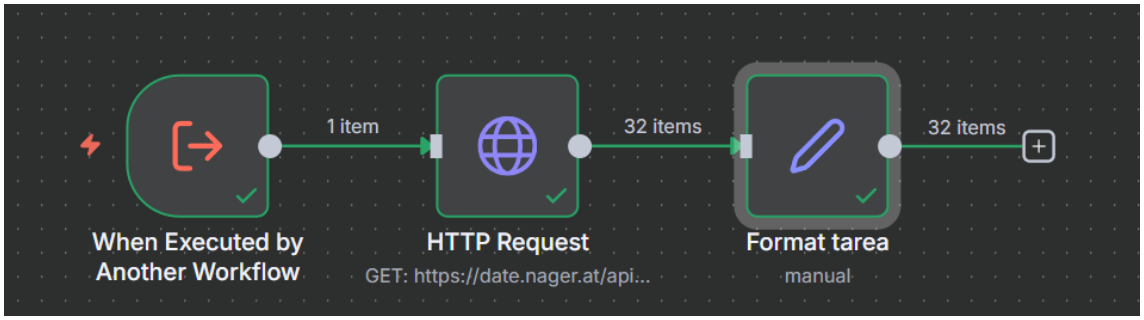
A continuación, añadimos el nodo HTTP Request, el cual, llama a la API de festivos de España para 2025 y obtiene una lista de todos los festivos.



Creamos un nodo Edit Fields para que el resultado siempre se devuelva bajo la variable tarea, al flujo principal, independientemente del sub-flujo, creamos un campo tarea y le asignamos el valor del array de los nombres locales de las fiestas, obtenido en el nodo anterior. Por lo que, el valor final de tarea será una lista con todas las festividades españolas.

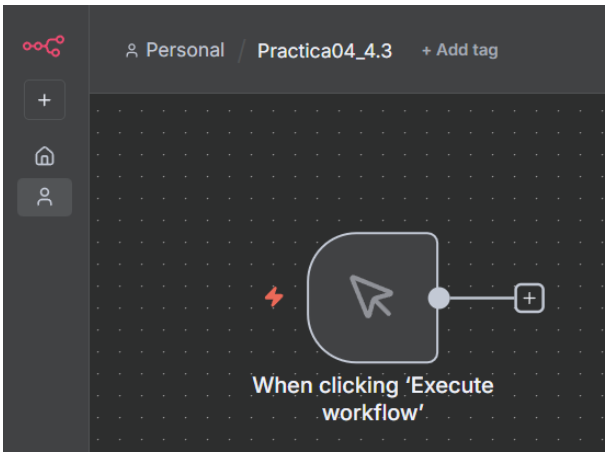


Observamos la estructura final de nuestro workflow:



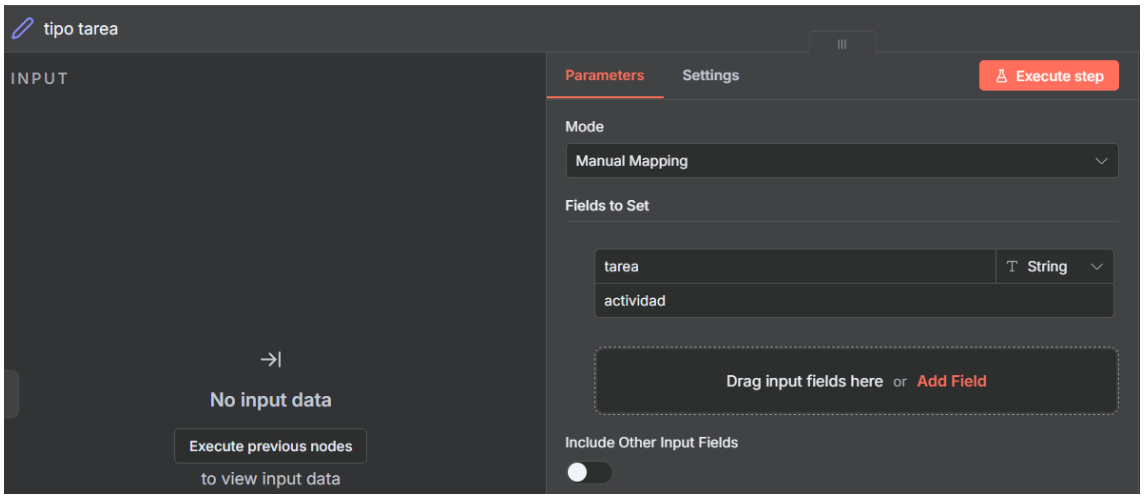
2. *Cree un flujo principal que comience con un Manual Trigger.*

Creamos el flujo principal "Practica04_4.3". Comienza con un "Manual Trigger".



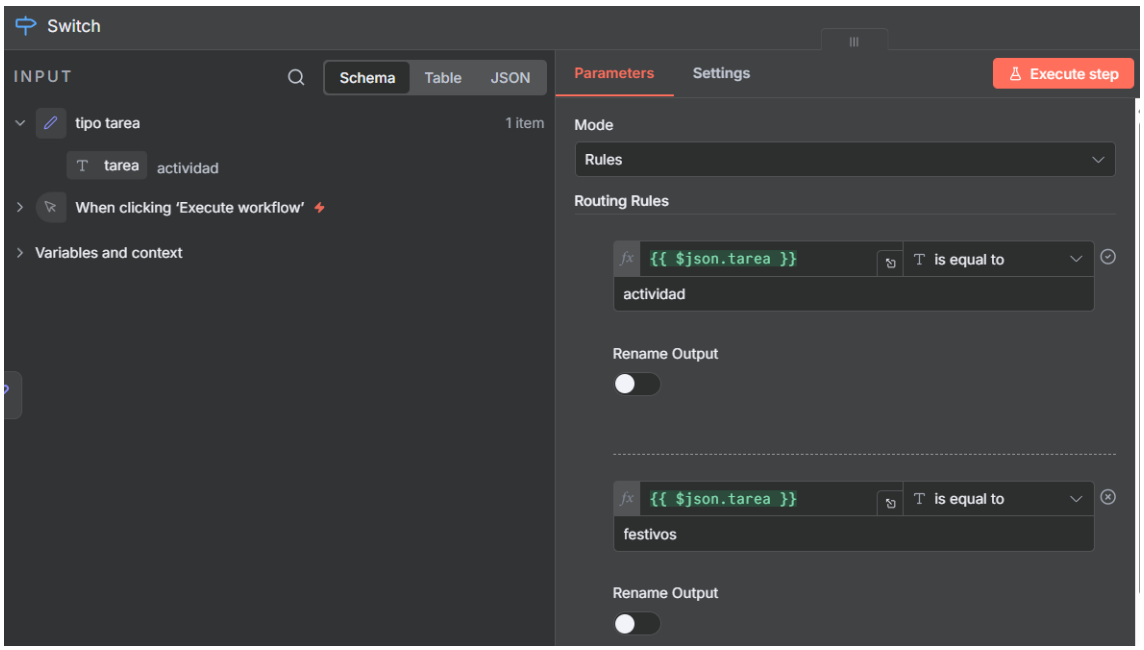
3. *Añada un nodo Edit Fields (Set) que cree una variable de control, por ejemplo, {"tarea": "actividad"}.*

Añadimos un nodo "Edit Fields (Set)" que crea una variable de control. Establecemos el campo tarea con el valor de cadena "actividad".

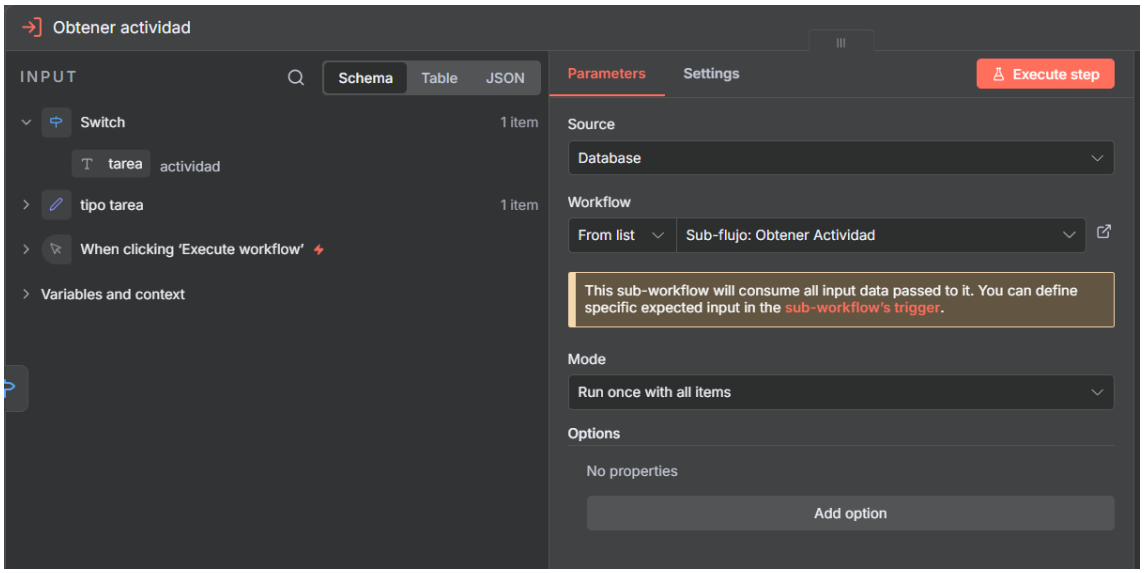


4. *Use un nodo Switch que evalúe el campo tarea.*
    - *Si tarea es "actividad", conecte un nodo Execute Workflow que llame al sub-flujo "Obtener Actividad".*

Conectamos un nodo "Switch" que evalúa si el contenido de la variable tarea es actividad o festivos.

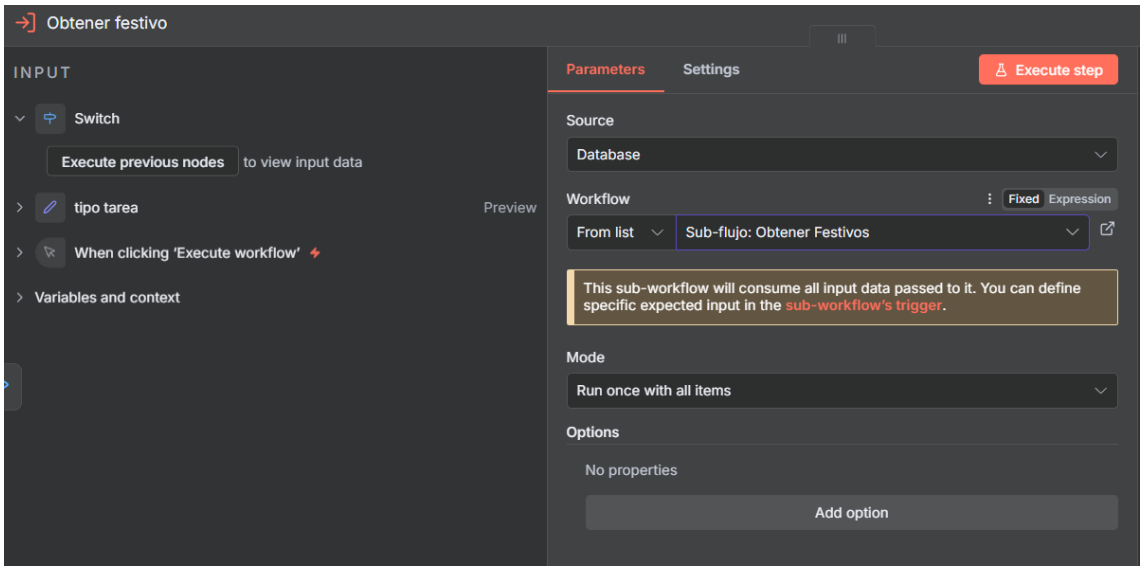


En la rama donde se da el caso de que el campo tarea sea “actividad", conectamos un nodo "Execute Workflow" ("Obtener actividad") que llama a nuestro "Sub-flujo: Obtener Actividad".



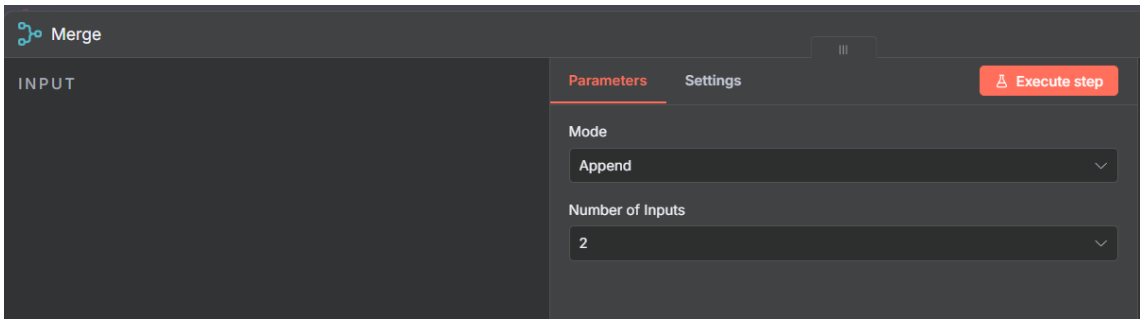
- *Si tarea es "festivos", conecte un nodo Execute Workflow que llame al sub-flujo "Obtener Festivos".*

En la rama donde se da el caso de que el campo tarea sea “festivos", conectamos un nodo "Execute Workflow" ("Obtener festivos") que llama a nuestro "Sub-flujo: Obtener Festivos".



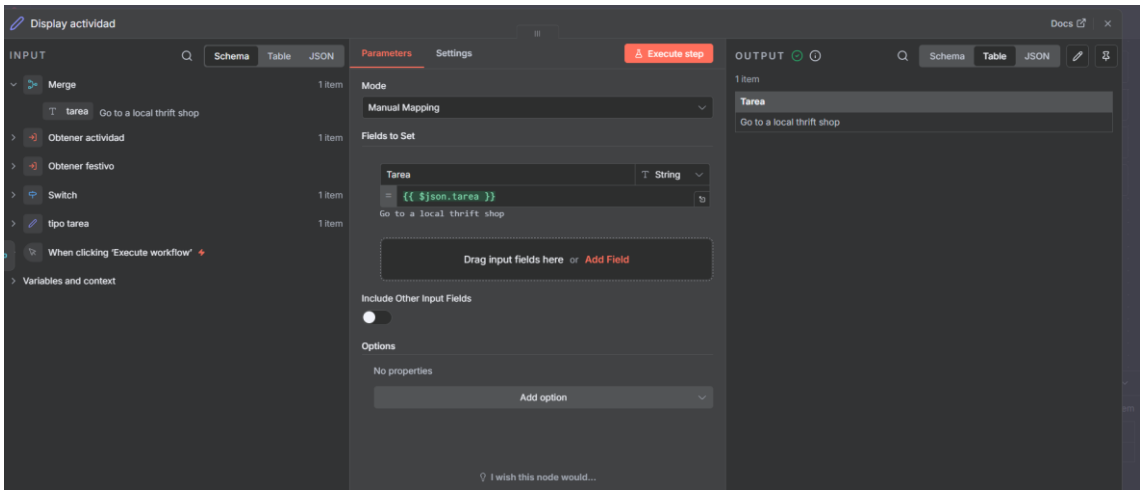
5. *Conecte un nodo Merge a las salidas de ambos nodos Execute Workflow para consolidar el resultado.*

Conectamos las salidas de ambos nodos "Execute Workflow" a un nodo "Merge". Lo configuramos en modo "Append" y "Number of Inputs" en 2. Esto unificará el resultado, sin importar qué rama se haya ejecutado.

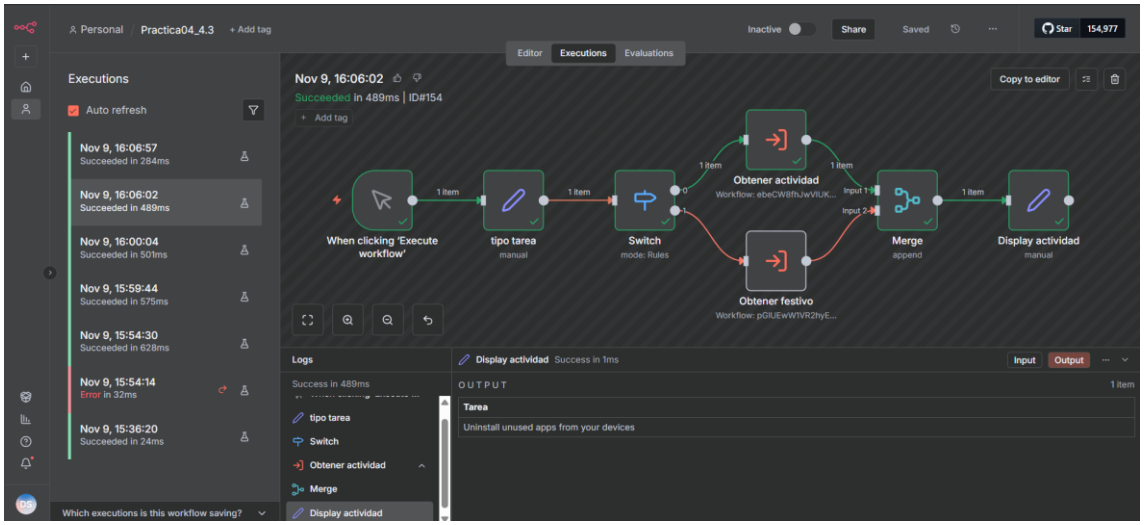


6. *Añada un nodo Edit Fields (Set) final para verificar la salida. Pruebe el flujo cambiando el valor de la variable tarea a "festivos" y vuelva a ejecutarlo para confirmar que se llama al sub-flujo correcto.*

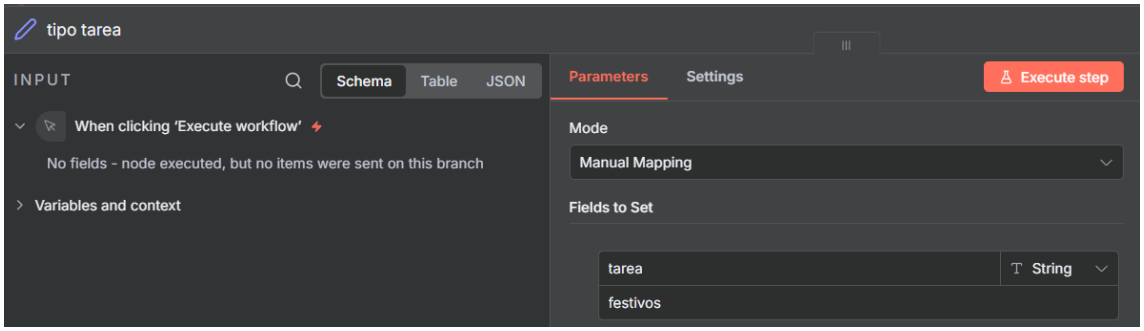
Después del "Merge", añadimos un nodo "Edit Fields (Set)" ("Display actividad"). Su único propósito es tomar el campo tarea (que ahora contiene el resultado estandarizado del sub-flujo que se haya ejecutado) y mostrarlo para verificar el resultado.



Ejecutamos el workflow. El nodo "tipo tarea" establece tarea en "actividad". El "Switch" envía la ejecución por la rama superior. Se llama al "Sub-flujo: Obtener Actividad". El "Merge" recibe el resultado, y el "Display actividad" muestra una sola actividad aleatoria. La ejecución es exitosa.



Ahora, modificamos el nodo "tipo tarea". Cambiamos el valor del campo tarea de "actividad" a "festivos".



Volvemos a ejecutar el mismo workflow. Esta vez, el "Switch" envía la ejecución por la rama inferior. Se llama al "Sub-flujo: Obtener Festivos", que devuelve 32 ítems. El "Merge" recibe los 32 ítems, y el "Display actividad" muestra la lista de todos los festivos ("Año Nuevo", "Día de Reyes / Epifanía...", etc.). Esto demuestra que el flujo orquestador funciona, seleccionando dinámicamente qué sub-flujo ejecutar basándose en una variable.