

# Memoria Práctica 7

**Jaime Parra Jiménez**

INTEGRACIÓN DE TECNOLOGÍAS Y SERVICIOS INFORMÁTICOS

Universidad de Almería

Correo: [jj451@inlumine.ual.es](mailto:jj451@inlumine.ual.es)

1 de diciembre de 2025

# Índice general

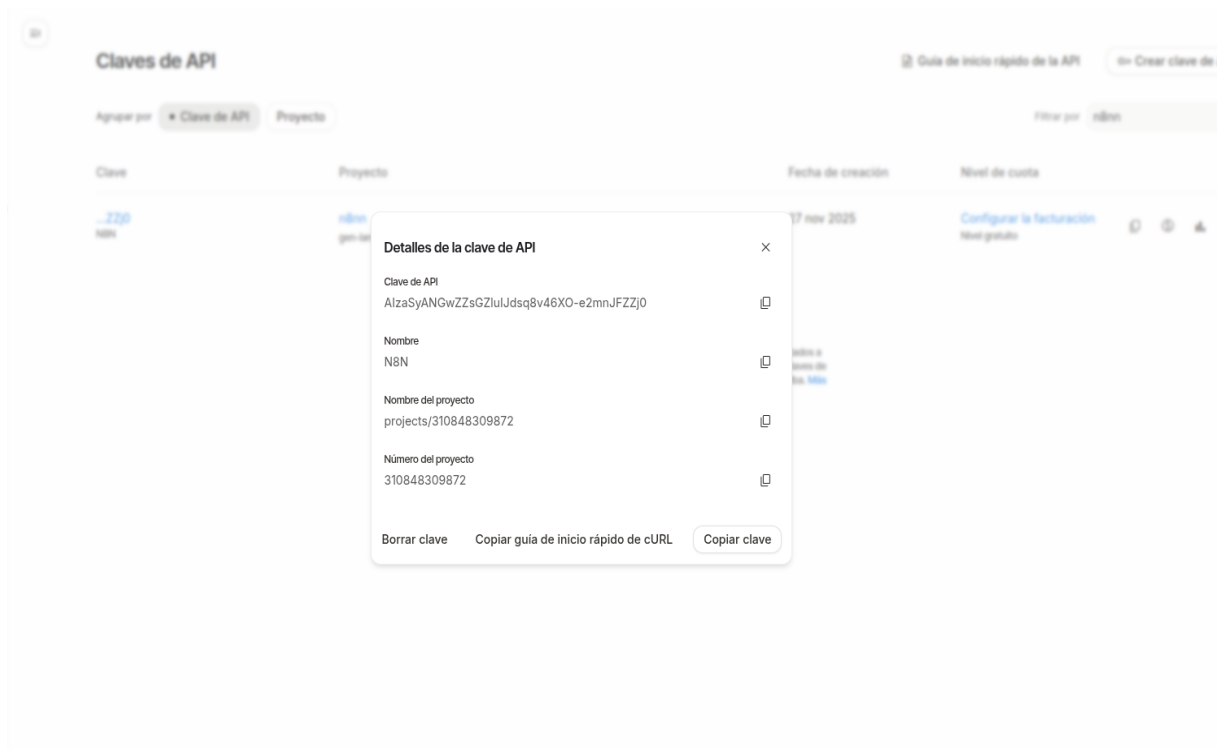
1. Ejercicio Guiado 1	2
2. Ejercicio 1	8
3. Ejercicio 2	11
4. Ejercicio 3	14

# Capítulo 1

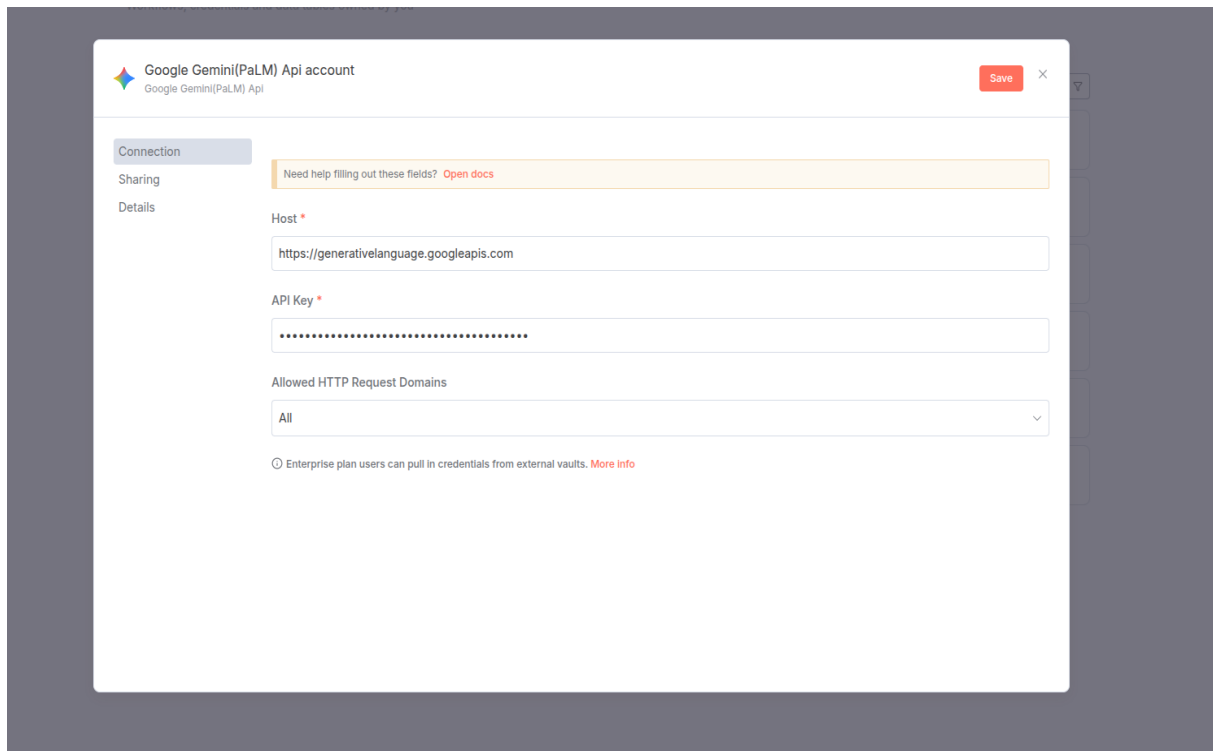
## Ejercicio Guiado 1

El objetivo del ejercicio guiado es conectar la IA a nuestra base de datos de tareas de la Práctica 6. Crear un flujo que lea tareas sin analizar, usar Gemini para determinar su prioridad basándose en la descripción, y actualizar la base de datos.

El primer paso es acceder a Google AI para obtener una nueva clave de la API.



Con la clave ya generada, en n8n se añadirá una nueva credencial de Google Gemini API donde se copiará la API key generada.

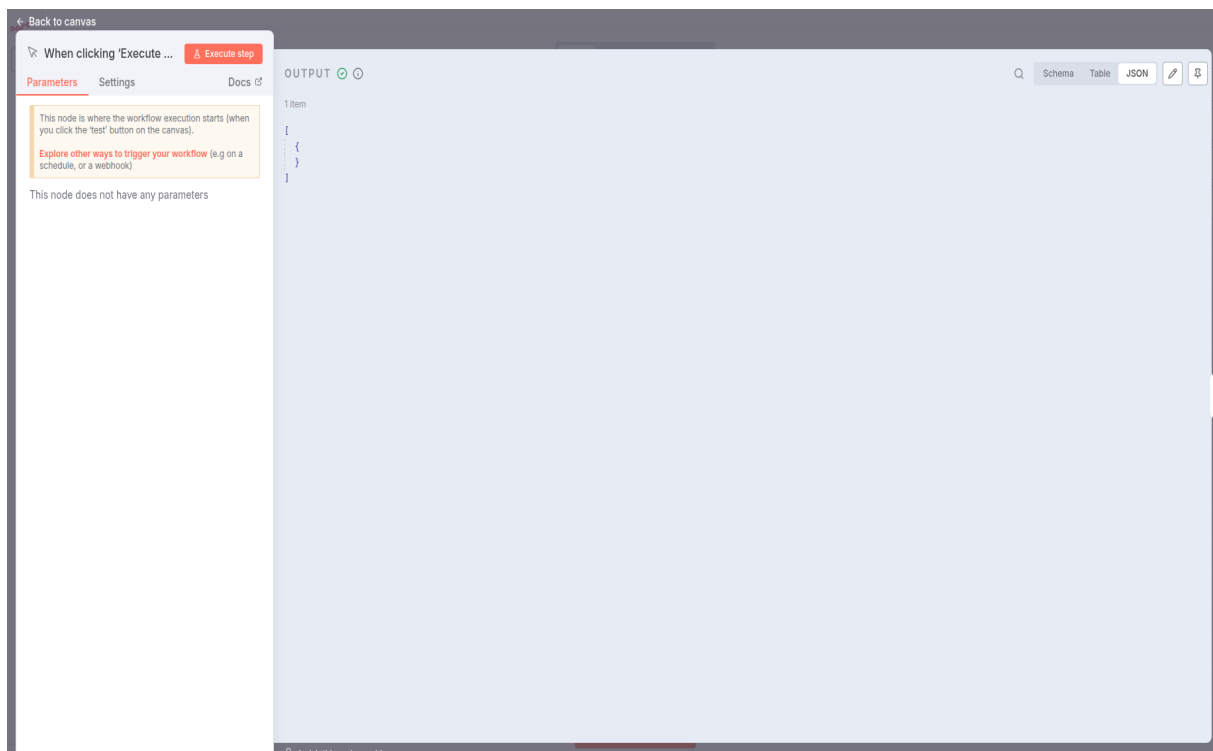


The screenshot shows the 'Google Gemini(PaLM) Api account' configuration window in n8n. The window has a title bar with the Google Gemini logo and the text 'Google Gemini(PaLM) Api account'. Below the title bar, there are three tabs: 'Connection', 'Sharing', and 'Details'. The 'Connection' tab is selected. The main content area of the 'Connection' tab contains the following fields:

- A message: 'Need help filling out these fields? [Open docs](#)'
- 'Host \*': A text input field containing 'https://generativelanguage.googleapis.com'.
- 'API Key \*': A text input field with a masked key represented by dots.
- 'Allowed HTTP Request Domains': A dropdown menu with 'All' selected.
- A footer note: 'Enterprise plan users can pull in credentials from external vaults. [More info](#)'

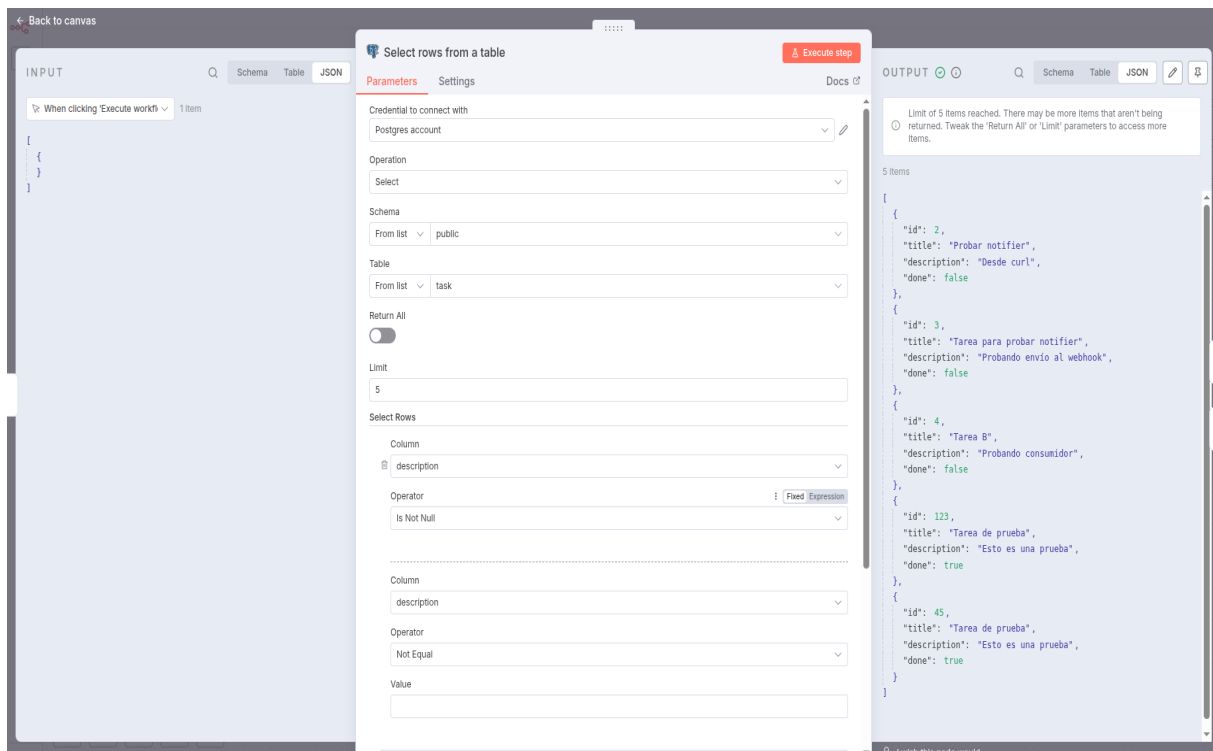
A red 'Save' button is located in the top right corner of the window.

A continuación, se crea el flujo dde trabajo. Primero se coloca un nodo Manual Trigger.

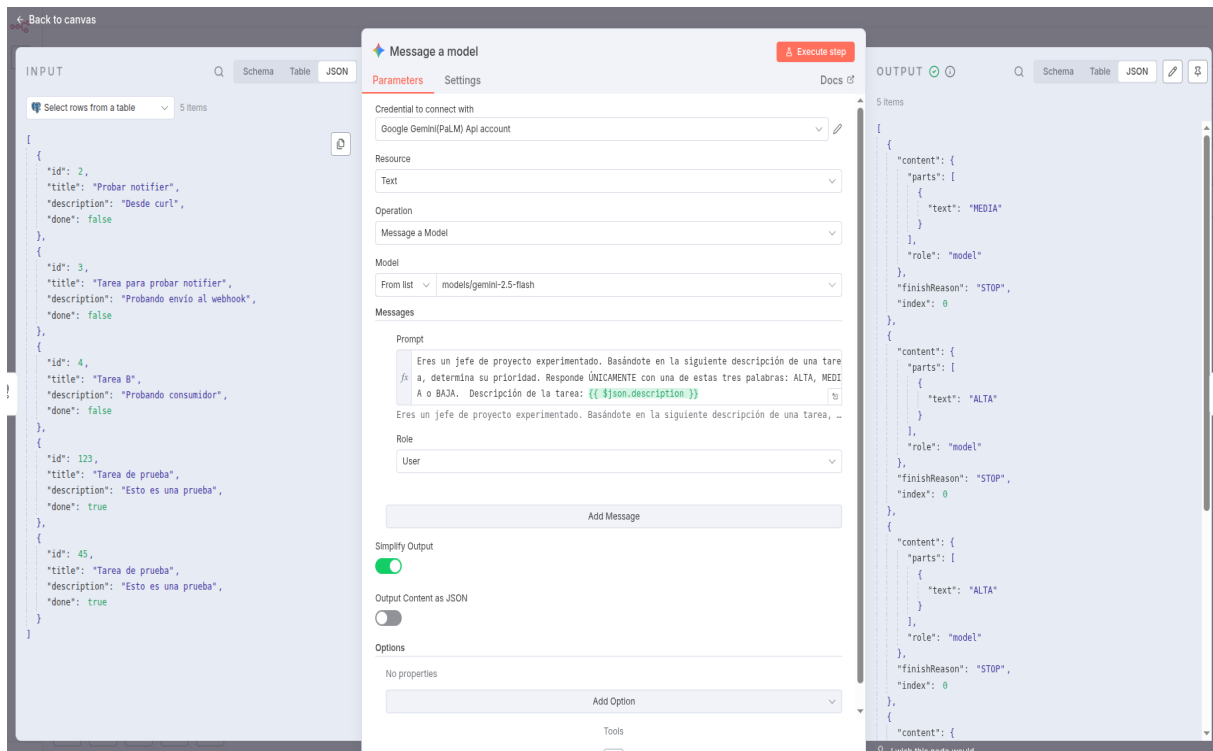


The screenshot shows the n8n workflow editor. On the left, there is a sidebar with a 'Back to canvas' link at the top. Below it, there is a node titled 'When clicking \'Execute ...\'' with a red 'Execute step' button. The node has three tabs: 'Parameters', 'Settings', and 'Docs'. The 'Parameters' tab is selected, and it shows a message: 'This node is where the workflow execution starts (when you click the \'test\' button on the canvas). Explore other ways to trigger your workflow (e.g on a schedule, or a webhook)'. Below this, it says 'This node does not have any parameters'. The main canvas area on the right is titled 'OUTPUT' and shows a JSON output: '1 item' followed by an empty object '{ }'. At the bottom of the canvas, there is a small red bar with the text 'I wish this node would...'.

A este se le añade un nodo PostgreSQL para obtener las tareas de la base da datos task, que previamente debe estar iniciada.



Añadir un nodo Google Gemini al PostgreSQL para analizar las tareas. Se usará un prompt para que Gemini defina la prioridad de las tareas.



Se añadirá un nodo merge para unificar la salida de Gemini con la de PostgreSQL.

The screenshot shows the 'Merge' node configuration in a workflow editor. The 'INPUT' panel on the left displays a JSON array of 5 items. The 'Merge' node's 'Parameters' tab is active, showing 'Mode' set to 'Combine', 'Combine By' set to 'Position', and 'Number of Inputs' set to '2'. The 'OUTPUT' panel on the right shows the resulting JSON array, which is a flattened version of the input data.

```
INPUT: [
  {
    "id": 2,
    "title": "Probar notifier",
    "description": "Desde curl",
    "done": false
  },
  {
    "id": 3,
    "title": "Tarea para probar notifier",
    "description": "Probando envío al webhook",
    "done": false
  },
  {
    "id": 4,
    "title": "Tarea B",
    "description": "Probando consumidor",
    "done": false
  },
  {
    "id": 123,
    "title": "Tarea de prueba",
    "description": "Esto es una prueba",
    "done": true
  },
  {
    "id": 45,
    "title": "Tarea de prueba",
    "description": "Esto es una prueba",
    "done": true
  }
]
```

Merge Parameters:

- Mode: Combine
- Combine By: Position
- Number of Inputs: 2
- Options: No properties

```
OUTPUT: [
  {
    "id": 2,
    "title": "Probar notifier",
    "description": "Desde curl",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "MEDIA"
        }
      ],
      "role": "model"
    },
    "finishReason": "STOP",
    "index": 0
  },
  {
    "id": 3,
    "title": "Tarea para probar notifier",
    "description": "Probando envío al webhook",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "ALTA"
        }
      ],
      "role": "model"
    },
    "finishReason": "STOP",
    "index": 0
  },
  {
    "id": 4,
    "title": "Tarea B",
    "description": "Probando consumidor",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "ALTA"
        }
      ],
      "role": "model"
    },
    "finishReason": "STOP",
    "index": 0
  },
  {
    "id": 123,
    "title": "Tarea de prueba",
    "description": "Esto es una prueba",
    "done": true
  },
  {
    "id": 45,
    "title": "Tarea de prueba",
    "description": "Esto es una prueba",
    "done": true
  }
]
```

Con un nodo Edit Fields (Set) se filtrarán los campos de ID, title y prioridad de cada tarea.

The screenshot shows the 'Edit Fields (Set)' node configuration in a workflow editor. The 'INPUT' panel on the left displays a JSON array of 5 items. The 'Edit Fields (Set)' node's 'Parameters' tab is active, showing 'Mode' set to 'Manual Mapping'. The 'Fields to Set' section shows three fields being mapped: 'ID' to '{{\$json.id}}', 'title' to '{{\$json.title}}', and 'Prioridad' to '{{\$json.content.parts[0].text}}'. The 'OUTPUT' panel on the right shows the resulting JSON array, where the fields are filtered and renamed.

```
INPUT: [
  {
    "id": 2,
    "title": "Probar notifier",
    "description": "Desde curl",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "MEDIA"
        }
      ],
      "role": "model"
    },
    "finishReason": "STOP",
    "index": 0
  },
  {
    "id": 3,
    "title": "Tarea para probar notifier",
    "description": "Probando envío al webhook",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "ALTA"
        }
      ],
      "role": "model"
    },
    "finishReason": "STOP",
    "index": 0
  },
  {
    "id": 4,
    "title": "Tarea B",
    "description": "Probando consumidor",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "ALTA"
        }
      ],
      "role": "model"
    },
    "finishReason": "STOP",
    "index": 0
  },
  {
    "id": 123,
    "title": "Tarea de prueba",
    "description": "Esto es una prueba",
    "done": true
  },
  {
    "id": 45,
    "title": "Tarea de prueba",
    "description": "Esto es una prueba",
    "done": true
  }
]
```

Edit Fields (Set) Parameters:

- Mode: Manual Mapping
- Fields to Set:
  - ID: String, Value: {{\$json.id}}
  - title: String, Value: {{\$json.title}}
  - Prioridad: String, Value: {{\$json.content.parts[0].text}}
- Include Other Input Fields: ☐
- Options: No properties

```
OUTPUT: [
  {
    "ID": "2",
    "title": "Probar notifier",
    "Prioridad": "MEDIA"
  },
  {
    "ID": "3",
    "title": "Tarea para probar notifier",
    "Prioridad": "ALTA"
  },
  {
    "ID": "4",
    "title": "Tarea B",
    "Prioridad": "ALTA"
  },
  {
    "ID": "123",
    "title": "Tarea de prueba",
    "Prioridad": "BAJA"
  },
  {
    "ID": "45",
    "title": "Tarea de prueba",
    "Prioridad": "BAJA"
  }
]
```

El siguiente paso es conectar un nodo Switch para separar las tareas según su prioridad.

The screenshot shows the Node-RED interface with a Switch node configured. The input JSON is as follows:

```
[{"ID": "2", "title": "Probar notificar", "Prioridad": "MEDIA"}, {"ID": "3", "title": "Tarea para probar notificar", "Prioridad": "ALTA"}, {"ID": "4", "title": "Tarea B", "Prioridad": "ALTA"}, {"ID": "123", "title": "Tarea de prueba", "Prioridad": "BAJA"}, {"ID": "45", "title": "Tarea de prueba", "Prioridad": "BAJA"}]
```

The Switch node is configured with the following routing rules:

- Rule 1: `js {{{ $json.Prioridad }}} ALTA`
- Rule 2: `js {{{ $json.Prioridad }}} MEDIA`
- Rule 3: `value1 value2`

The output shows three streams of items based on these rules:

- Output 0 (2 items):

```
[{"ID": "3", "title": "Tarea para probar notificar", "Prioridad": "ALTA"}, {"ID": "4", "title": "Tarea B", "Prioridad": "ALTA"}]
```
- Output 1 (1 item):

```
[{"ID": "2", "title": "Probar notificar", "Prioridad": "MEDIA"}]
```
- Output 2 (2 items):

```
[{"ID": "123", "title": "Tarea de prueba", "Prioridad": "BAJA"}, {"ID": "45", "title": "Tarea de prueba", "Prioridad": "BAJA"}]
```

Se conectará un nodo Send Email a la salida 0, las tareas con prioridad alta, para mandar un aviso de que la tarea ha sido calificada de prioridad alta por la IA.

The screenshot shows the Node-RED interface with a Send Email node configured. The input JSON is as follows:

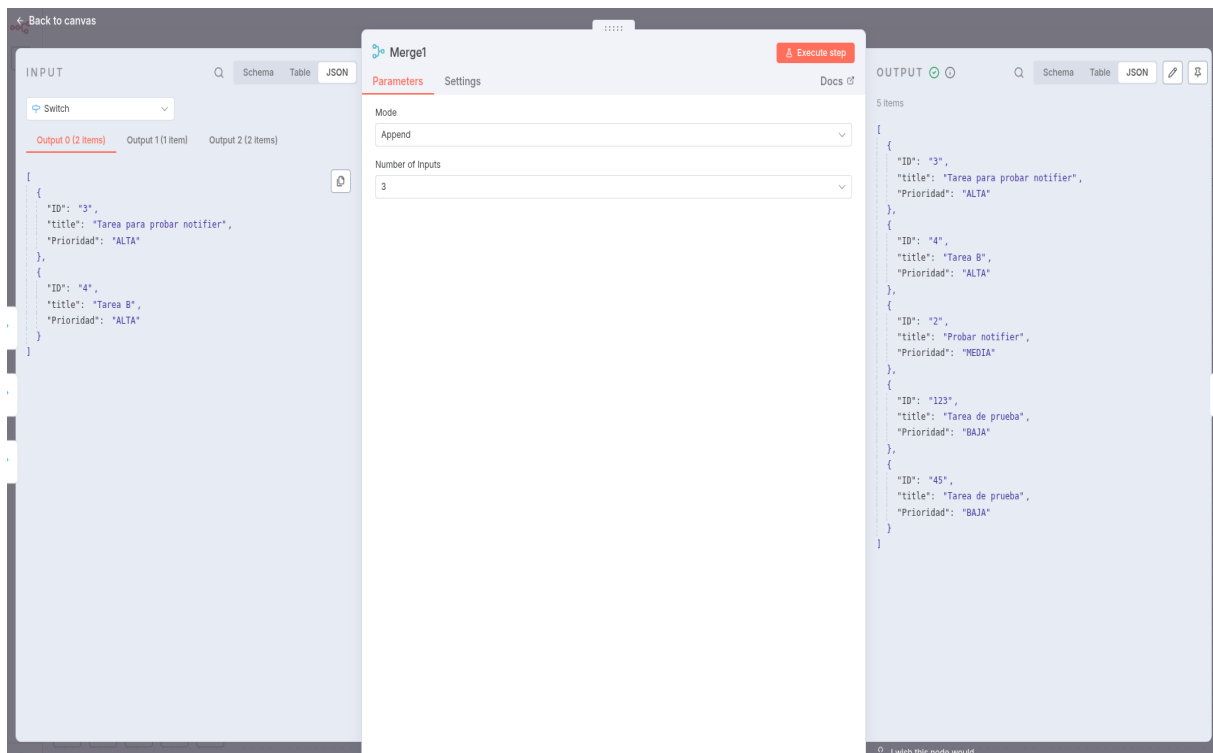
```
[{"ID": "3", "title": "Tarea para probar notificar", "Prioridad": "ALTA"}, {"ID": "4", "title": "Tarea B", "Prioridad": "ALTA"}]
```

The Send Email node is configured with the following parameters:

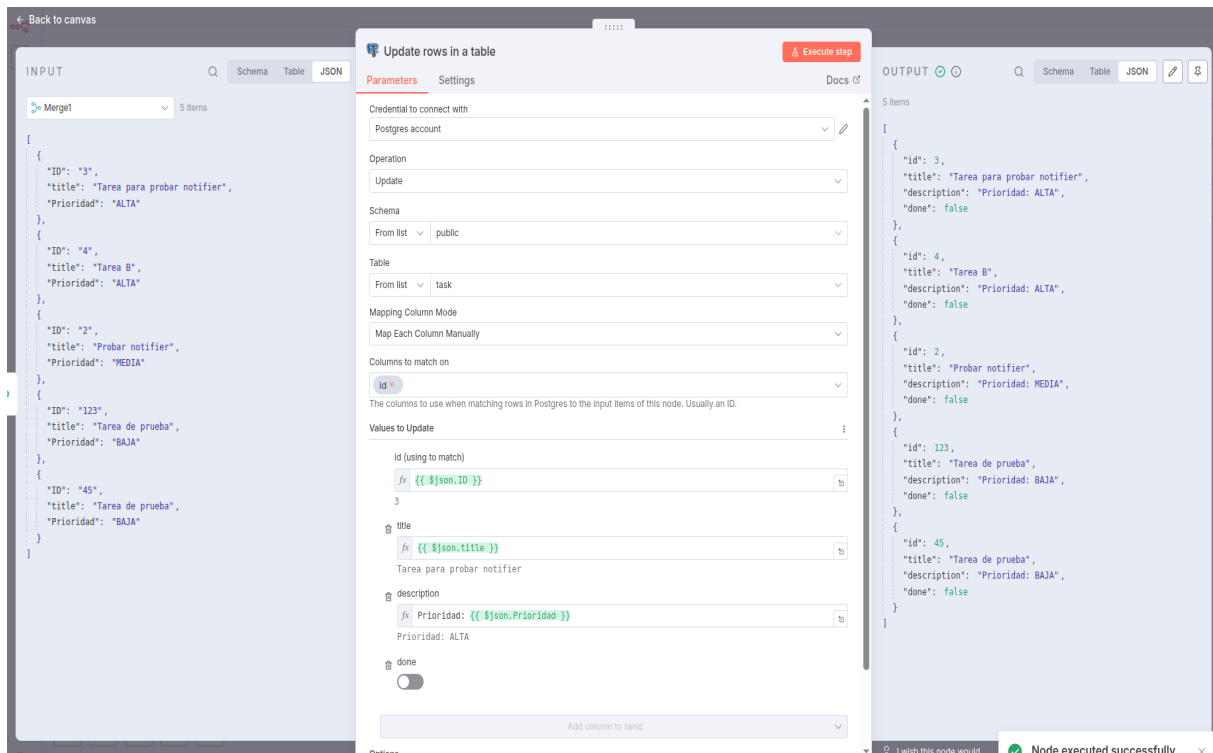
- Credential to connect with: SMTP account
- Operation: Send
- From Email: `jpj451@inlumine.ual.es`
- To Email: `jpj451@inlumine.ual.es`
- Subject: `¡Alerta de Tarea Prioridad ALTA!`
- Email Format: HTML
- HTML: `La tarea "{{ $json.title }}" (ID: "{{ $json.ID }}") ha sido clasificada como ALTA prioridad por la IA.`

The output shows the email being sent successfully, with a status message: "Node executed successfully".

A las salidas se le conectará un Merge para unificar las tareas de las tres salidas para poder actualizarlas en la base de datos.



Por último, con un nodo PostgreSQL se actualizará la base de datos indicando la prioridad de cada tarea.

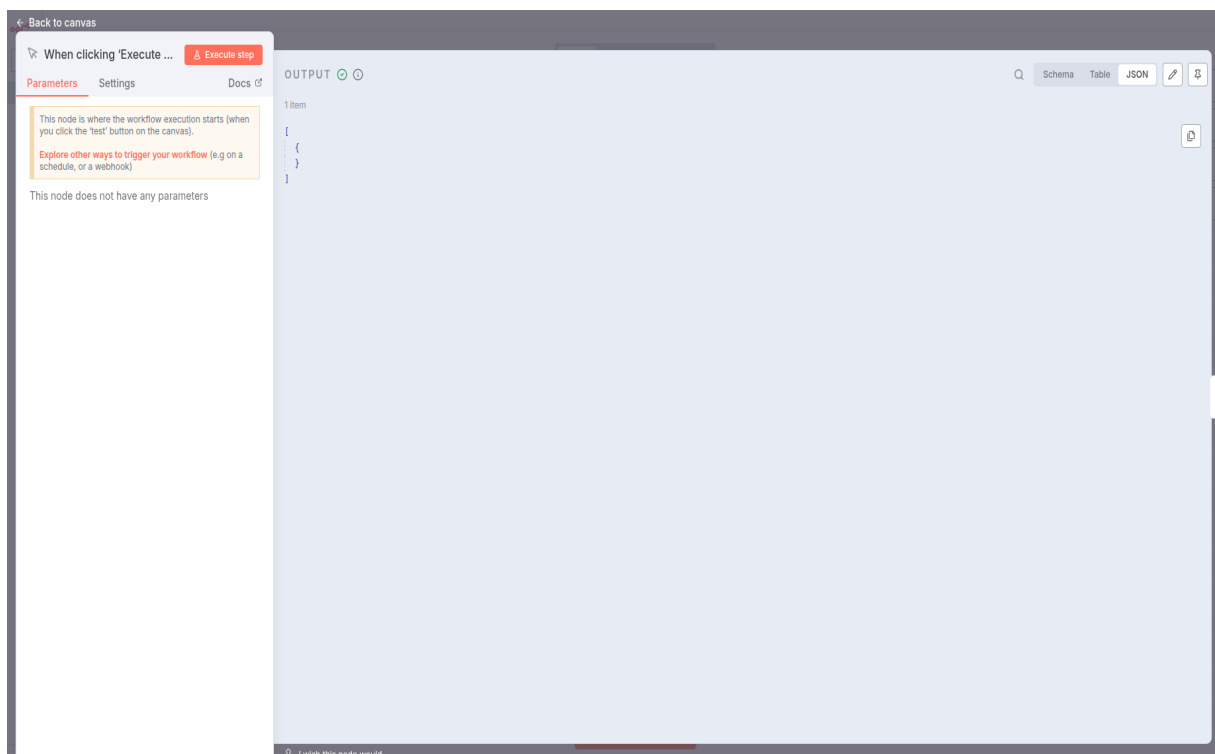




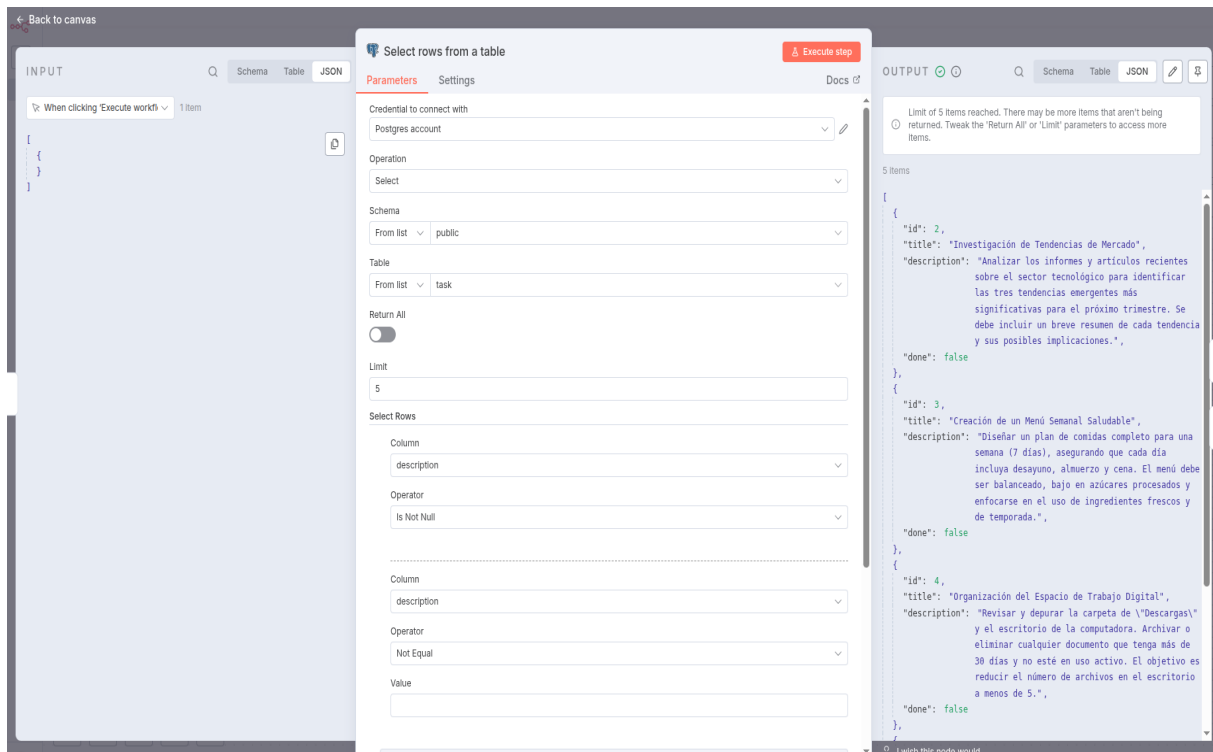
# Capítulo 2

## Ejercicio 1

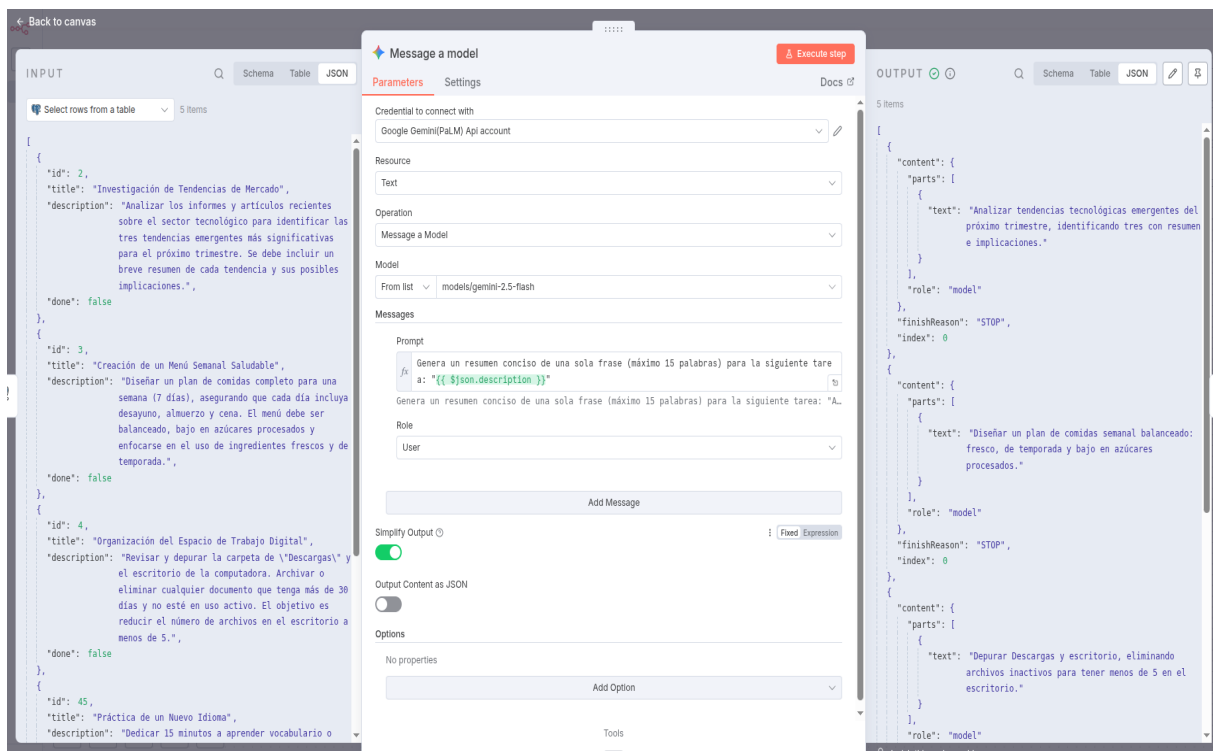
El objetivo del ejercicio 1 es modificar el flujo de trabajo guiado para que, en lugar de clasificar la prioridad, genere un resumen de la tarea. Para empezar con el flujo de trabajo se añade un nodo Manual Trigger.



A continuación se le conecta un nodo PostgreSQL para obtener las tareas de la base de datos.



El siguiente paso es modificar el prompt del ejercicio guiado para que en vez de definir la prioridad haga un resumen de la tarea de máximo 15 palabras.



A continuación se añade un nodo Merge para unificar el resumen a su respectiva tarea.

The screenshot shows the 'Merge' node configuration in a workflow editor. The 'INPUT' panel on the left displays a JSON array of 5 items. The 'Parameters' panel in the center shows the 'Mode' set to 'Combine', 'Combine By' set to 'Position', and 'Number of Inputs' set to 2. The 'OUTPUT' panel on the right shows the resulting JSON array, which is a single object containing the merged data from the input items.

```
INPUT
[
  {
    "id": 2,
    "title": "Investigación de Tendencias de Mercado",
    "description": "Analizar los informes y artículos recientes sobre el sector tecnológico para identificar las tres tendencias emergentes más significativas para el próximo trimestre. Se debe incluir un breve resumen de cada tendencia y sus posibles implicaciones.",
    "done": false
  },
  {
    "id": 3,
    "title": "Creación de un Menú Semanal Saludable",
    "description": "Diseñar un plan de comidas completo para una semana (7 días), asegurando que cada día incluya desayuno, almuerzo y cena. El menú debe ser balanceado, bajo en azúcares procesados y enfocarse en el uso de ingredientes frescos y de temporada.",
    "done": false
  },
  {
    "id": 4,
    "title": "Organización del Espacio de Trabajo Digital",
    "description": "Revisar y depurar la carpeta de 'Descargas' y el escritorio de la computadora. Archivar o eliminar cualquier documento que tenga más de 30 días y no esté en uso activo. El objetivo es reducir el número de archivos en el escritorio a menos de 5.",
    "done": false
  },
  {
    "id": 45,
    "title": "Práctica de un Nuevo Idioma",
    "description": "Dedicar 15 minutos a aprender vocabulario o gramática de un idioma nuevo con cualquier método."
  }
]

OUTPUT
[
  {
    "id": 2,
    "title": "Investigación de Tendencias de Mercado",
    "description": "Analizar los informes y artículos recientes sobre el sector tecnológico para identificar las tres tendencias emergentes más significativas para el próximo trimestre. Se debe incluir un breve resumen de cada tendencia y sus posibles implicaciones.",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "Analizar tendencias tecnológicas emergentes del próximo trimestre, identificando tres con resumen e implicaciones."
        }
      ]
    },
    "role": "model"
  },
  {
    "id": 3,
    "title": "Creación de un Menú Semanal Saludable",
    "description": "Diseñar un plan de comidas completo para una semana (7 días), asegurando que cada día incluya desayuno, almuerzo y cena. El menú debe ser balanceado, bajo en azúcares procesados y enfocarse en el uso de ingredientes frescos y de temporada.",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "Diseñar un plan de comidas semanal balanceado: fresco, de temporada y bajo en azúcares"
        }
      ]
    }
  }
]
```

Por último, se añade un Edit Fields (Set) para añadir un campo resumen donde se guardará la salida de Gemini.

The screenshot shows the 'Edit Fields (Set)' node configuration in a workflow editor. The 'INPUT' panel on the left displays a JSON array of 5 items. The 'Parameters' panel in the center shows the 'Mode' set to 'Manual Mapping'. The 'Fields to Set' section shows three fields being mapped: 'ID' to '{{\$json.id}}', 'title' to '{{\$json.title}}', and 'Resumen' to '{{\$json.content.parts[0].text}}'. The 'OUTPUT' panel on the right shows the resulting JSON array, which includes a 'Resumen' field for each item.

```
INPUT
[
  {
    "id": 2,
    "title": "Investigación de Tendencias de Mercado",
    "description": "Analizar los informes y artículos recientes sobre el sector tecnológico para identificar las tres tendencias emergentes más significativas para el próximo trimestre. Se debe incluir un breve resumen de cada tendencia y sus posibles implicaciones.",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "Analizar tendencias tecnológicas emergentes del próximo trimestre, identificando tres con resumen e implicaciones."
        }
      ]
    },
    "role": "model"
  },
  {
    "id": 3,
    "title": "Creación de un Menú Semanal Saludable",
    "description": "Diseñar un plan de comidas completo para una semana (7 días), asegurando que cada día incluya desayuno, almuerzo y cena. El menú debe ser balanceado, bajo en azúcares procesados y enfocarse en el uso de ingredientes frescos y de temporada.",
    "done": false,
    "content": {
      "parts": [
        {
          "text": "Diseñar un plan de comidas semanal balanceado: fresco, de temporada y bajo en azúcares"
        }
      ]
    }
  }
]

OUTPUT
[
  {
    "ID": "2",
    "title": "Investigación de Tendencias de Mercado",
    "Resumen": "Analizar tendencias tecnológicas emergentes del próximo trimestre, identificando tres con resumen e implicaciones."
  },
  {
    "ID": "3",
    "title": "Creación de un Menú Semanal Saludable",
    "Resumen": "Diseñar un plan de comidas semanal balanceado: fresco, de temporada y bajo en azúcares procesados."
  },
  {
    "ID": "4",
    "title": "Organización del Espacio de Trabajo Digital",
    "Resumen": "Depurar Descargas y escritorio, eliminando archivos inactivos para tener menos de 5 en el escritorio."
  },
  {
    "ID": "45",
    "title": "Práctica de un Nuevo Idioma",
    "Resumen": "Dedica 15 minutos a aprender vocabulario o gramática de un idioma nuevo con cualquier método."
  },
  {
    "ID": "123",
    "title": "Mantenimiento Básico de una Planta de Interior",
    "Resumen": "Inspeccionar, regar, limpiar hojas y revisar plagas de la planta."
  }
]
```

# Capítulo 3

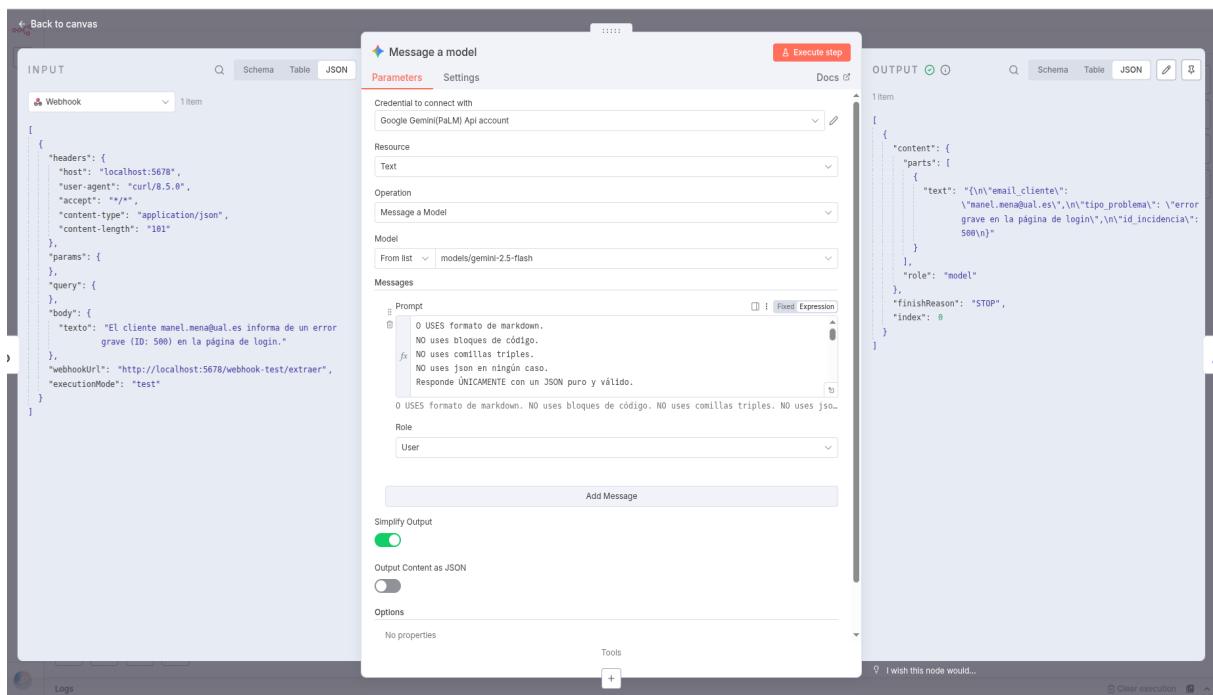
## Ejercicio 2

El objetivo del ejercicio 2 es crear un flujo que reciba texto no estructurado y extraiga datos en formato JSON, demostrando el patrón de "Prompt Fuerte". El primer paso es iniciar el flujo con un Webhook Trigger.

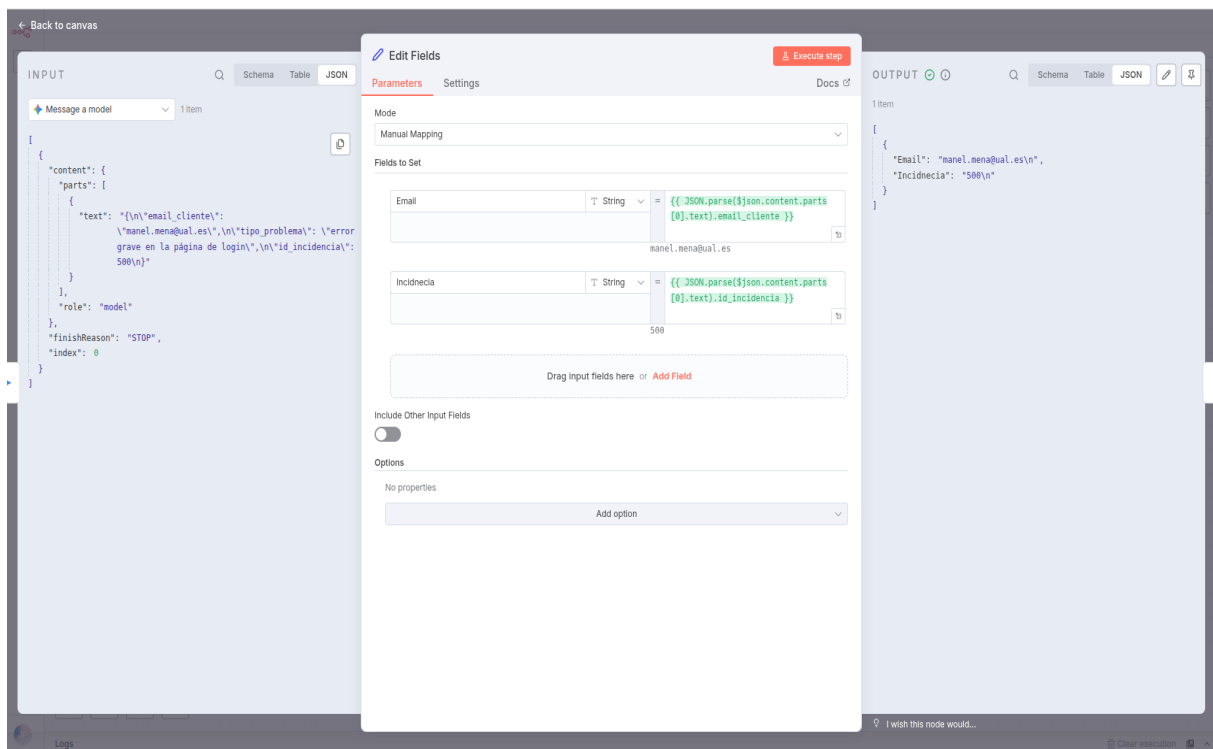
The screenshot shows the configuration of a Webhook trigger in a workflow tool. The central panel is titled 'Webhook' and has tabs for 'Parameters' and 'Settings'. Under 'Parameters', there are sections for 'Webhook URLs' (with 'Test URL' and 'Production URL' buttons), 'HTTP Method' (set to POST), 'Path' (set to 'extraer'), 'Authentication' (set to None), 'Respond' (set to Immediately), and 'Options' (set to No properties). A red button 'Listen for test event' is at the top right of the configuration panel. The right panel, titled 'OUTPUT', shows a JSON response with the following structure:

```
{
  "headers": {
    "host": "localhost:5678",
    "user-agent": "curl/8.5.0",
    "accept": "**/*",
    "content-type": "application/json",
    "content-length": "101"
  },
  "params": {
  },
  "query": {
  },
  "body": {
    "texto": "El cliente manel.mena@ual.es informa de un error grave (ID: 500) en la página de login."
  },
  "webhookUrl": "http://localhost:5678/webhook-test/extraer",
  "executionMode": "test"
}
```

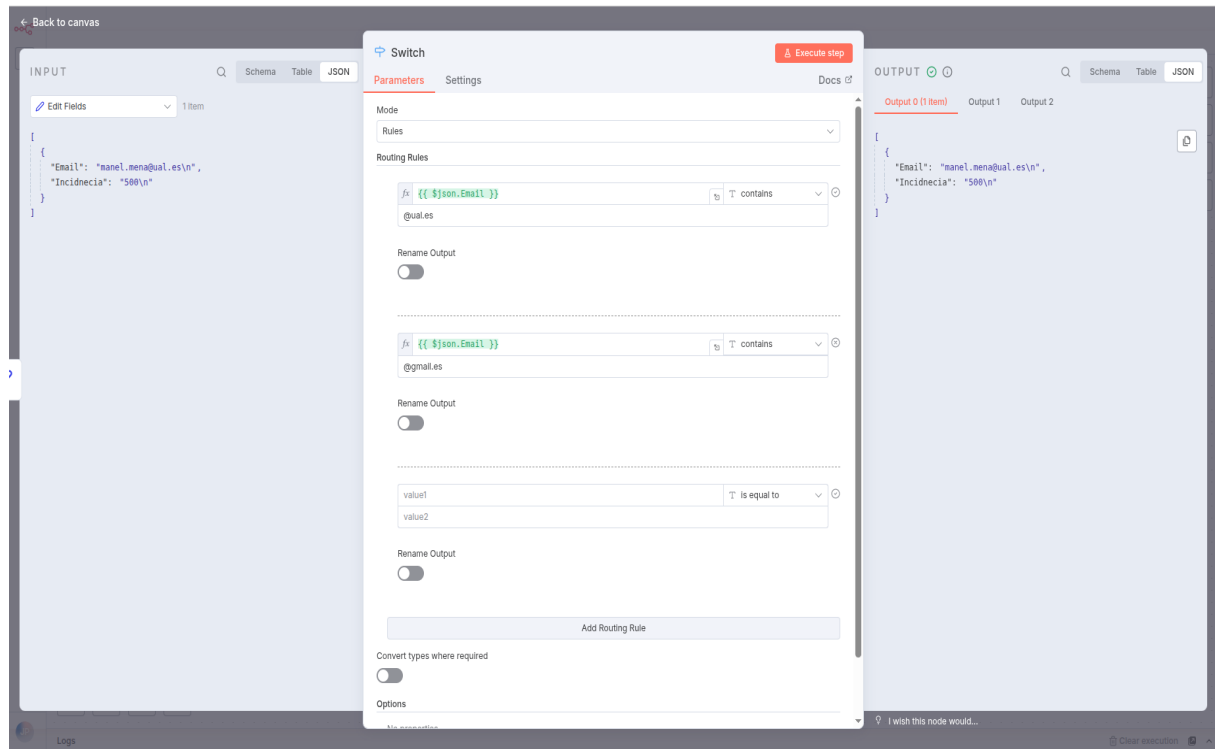
A continuación se añade un nodo Google Gemini donde se le pasará un prompt para que extraiga el email, el tipo de problema y la ID, y que responda únicamente en formato JSON.



Se le conecta un nodo Edit Fields (Set) para crear los campos Email e Incidencia partir del JSON.



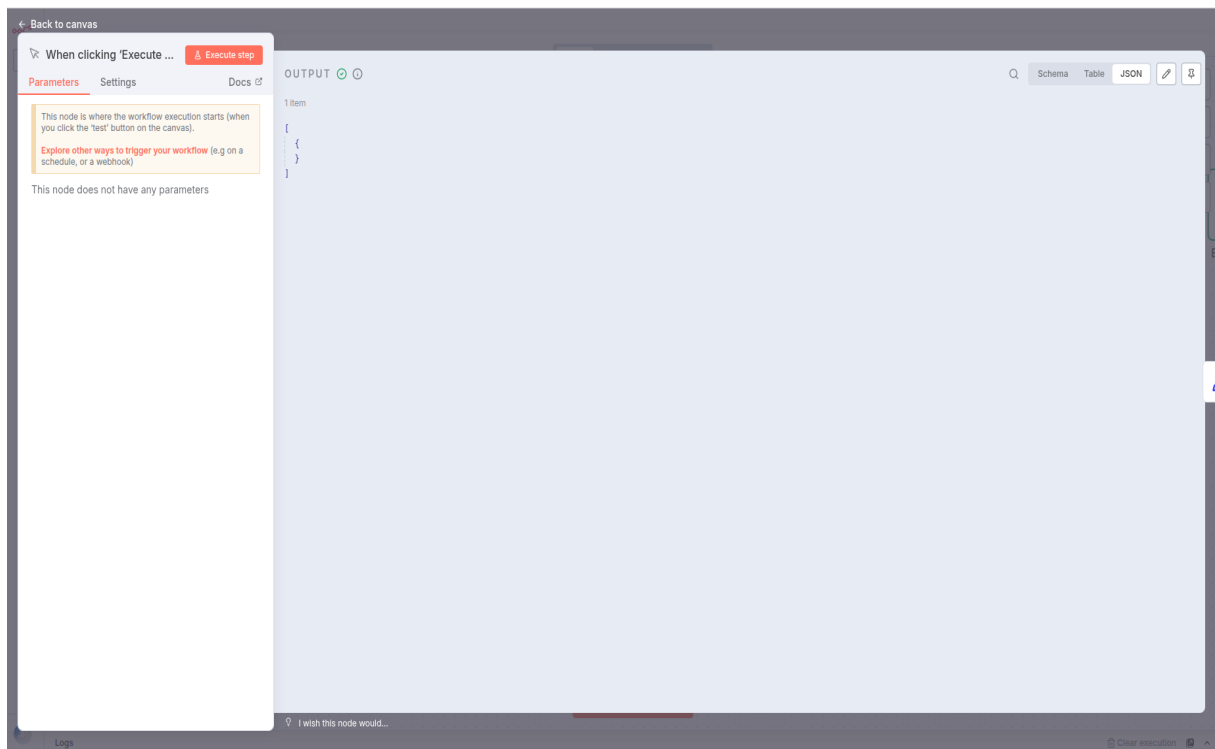
Por último, se añade un nodo Switch para enrutar el flujo basándose en el campo email.



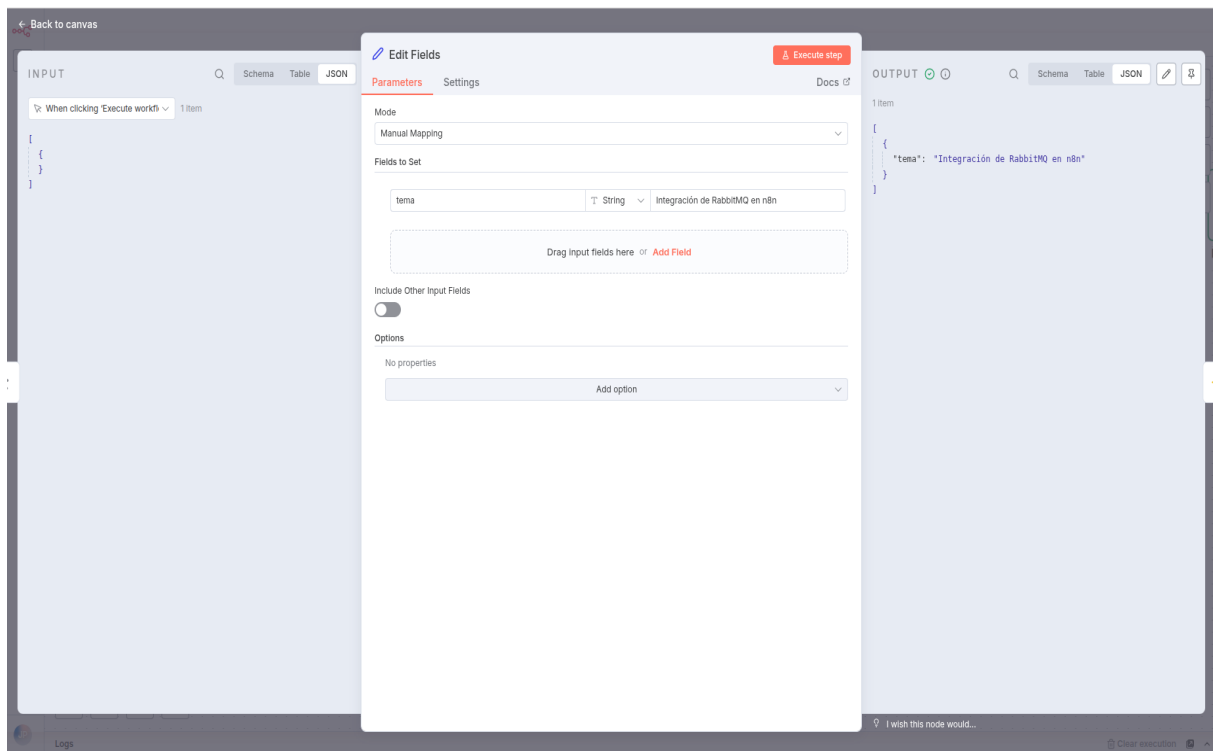
# Capítulo 4

## Ejercicio 3

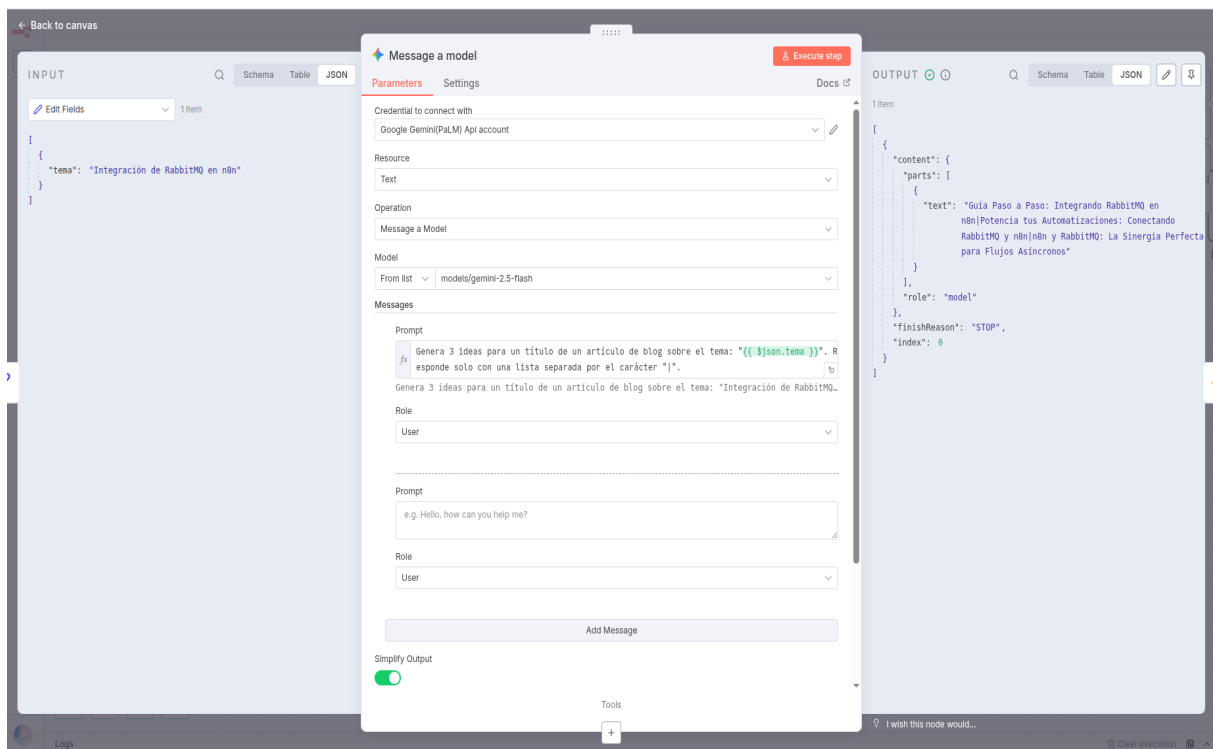
El objetivo del ejercicio 3 es crear un flujo donde la salida de una llamada a la IA se utiliza como entrada para una segunda llamada a la IA. Para empezar el flujo de trabajo se añade un Manual Trigger.



El siguiente paso es añadir un nodo Edit Fields(Set) para definir el campo tema con el valor Integración de RabbitMQ en n8n.

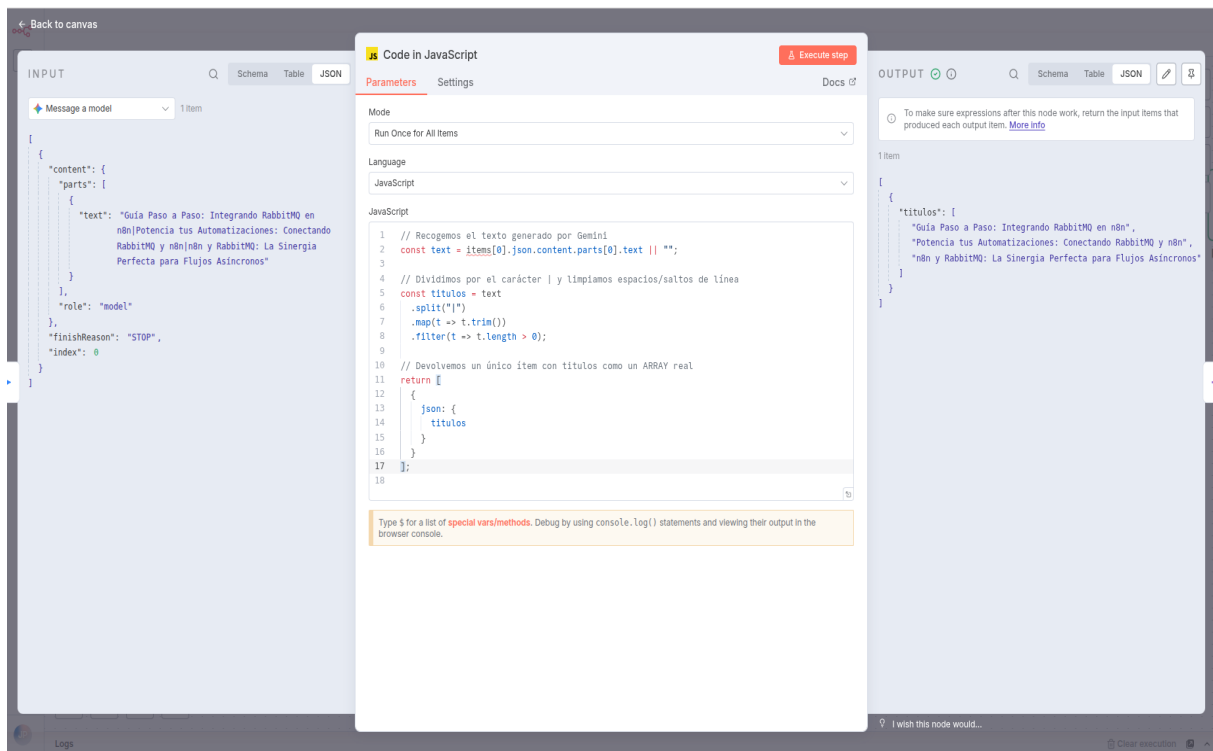


Al Edit Fields se le conecta un nodo Google Gemini para que cree tres titulos para un artículo de blog.

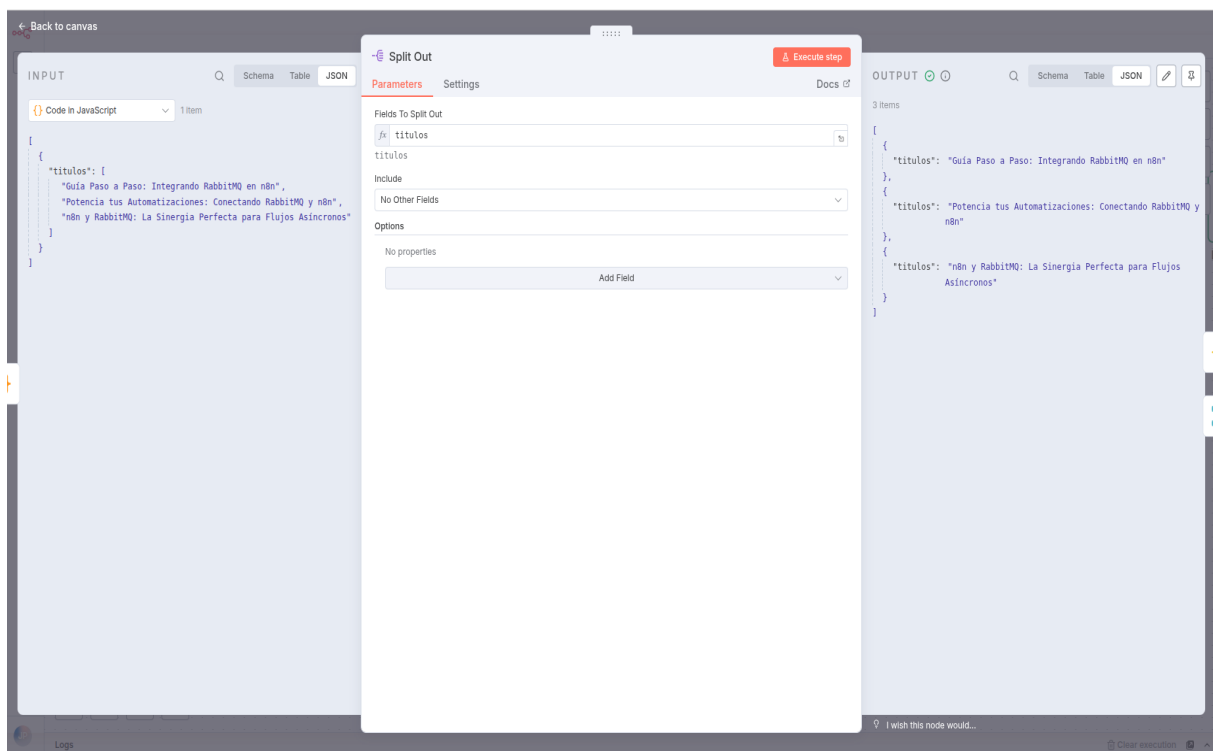




Para separar los tres títulos y darle formato de JSON se usa un nodo Code de Javascript.



A continuación se usa un nodo Split Out para dividir el array en ítems individuales.



Al Split Out se le conecta un segundo nodo Google Gemini para escribir un párrafo de unas 3 o 4 frases para cada título.

The screenshot shows the n8n workflow editor with the 'Message a model' node configured. The 'INPUT' panel on the left shows a 'Split Out' node with 3 items, each containing a JSON object with a 'titulos' field. The 'Message a model' node is configured with the following settings:

- Credential to connect with:** Google Gemini(PaLM) Api account
- Resource:** Text
- Operation:** Message a Model
- Model:** From list (models/gemini-2.5-flash)
- Prompt:**

```
Escribe un párrafo de introducción (3-4 frases) sin opciones ni nada, solamente una breve introducción para un artículo de blog con el siguiente título: "{{ $json.titulos }}" N o pongas nada de frases ni nada. simplemente la introducción sin nada mas
```
- Role:** User
- Simplify Output:** Enabled
- Output Content as JSON:** Disabled
- Options:** No properties

The 'OUTPUT' panel on the right shows the result of the node, which is a JSON object containing the generated text for each title.

Además, se añade un nodo Merge para recopilar los tres párrafos generados.

The screenshot shows the n8n workflow editor with the 'Merge' node configured. The 'INPUT' panel on the left shows the same 'Split Out' node with 3 items. The 'Merge' node is configured with the following settings:

- Mode:** Combine
- Combine By:** Position
- Number of Inputs:** 2
- Options:** No properties

The 'OUTPUT' panel on the right shows the result of the node, which is a JSON object containing the merged text for each title.

Por último, se añade un nodo Edits Fields(Set) para separar los campos de Título e Introducción.

The screenshot displays the n8n workflow editor with the 'Edit Fields' node configured in 'Manual Mapping' mode. The node is named 'Edit Fields1'. The 'Fields to Set' section shows two mappings: 'Titulo' mapped to '{{ \$json.titulos }}' and 'Introduccion' mapped to '{{ \$json.content.parts[0].text }}'. The 'Include Other Input Fields' toggle is turned off. The 'INPUT' panel shows a JSON array of two items, and the 'OUTPUT' panel shows the resulting JSON array after the mapping.

**INPUT**

```
[{"titulos": "Guía Paso a Paso: Integrando RabbitMQ en n8n", "content": {"parts": [{"text": "En el dinámico mundo de la automatización, la gestión eficiente y fiable de los datos es clave para construir sistemas robustos. RabbitMQ, un potente broker de mensajes, se erige como una solución ideal para asegurar una comunicación asíncrona y resistente entre tus servicios. Cuando lo combinamos con la versatilidad de n8n, una herramienta de automatización de código abierto, las posibilidades de crear flujos de trabajo escalables y a prueba de fallos se multiplican. Esta guía te mostrará cómo integrar RabbitMQ en tus automatizaciones de n8n, paso a paso, para llevar tus proyectos al siguiente nivel."}]}, "role": "model"}, {"titulos": "Potencia tus Automatizaciones: Conectando RabbitMQ y n8n", "content": {"parts": [{"text": "En el dinámico mundo de la automatización, optimizar la comunicación entre tus sistemas y servicios es crucial para la eficiencia. Este artículo explora cómo la potente combinación de RabbitMQ, un robusto broker de mensajes, y n8n, una flexible herramienta de automatización, puede transformar tus flujos de trabajo. Descubre cómo integrar estas dos herramientas para construir automatizaciones más resilientes, escalables y capaces de manejar cualquier volumen de tareas. Prepárate para llevar tus procesos automatizados a un nivel superior de rendimiento y fiabilidad."}]}, "role": "model"}]
```

**OUTPUT**

```
[{"Titulo": "Guía Paso a Paso: Integrando RabbitMQ en n8n", "Introduccion": "En el dinámico mundo de la automatización, la gestión eficiente y fiable de los datos es clave para construir sistemas robustos. RabbitMQ, un potente broker de mensajes, se erige como una solución ideal para asegurar una comunicación asíncrona y resistente entre tus servicios. Cuando lo combinamos con la versatilidad de n8n, una herramienta de automatización de código abierto, las posibilidades de crear flujos de trabajo escalables y a prueba de fallos se multiplican. Esta guía te mostrará cómo integrar RabbitMQ en tus automatizaciones de n8n, paso a paso, para llevar tus proyectos al siguiente nivel."}, {"Titulo": "Potencia tus Automatizaciones: Conectando RabbitMQ y n8n", "Introduccion": "En el dinámico mundo de la automatización, optimizar la comunicación entre tus sistemas y servicios es crucial para la eficiencia. Este artículo explora cómo la potente combinación de RabbitMQ, un robusto broker de mensajes, y n8n, una flexible herramienta de automatización, puede transformar tus flujos de trabajo. Descubre cómo integrar estas dos herramientas para construir automatizaciones más resilientes, escalables y capaces de manejar cualquier volumen de tareas. Prepárate para llevar tus procesos automatizados a un nivel superior de rendimiento y fiabilidad."}]
```