

# Memoria Práctica 1

**Jaime Parra Jiménez**

INTEGRACIÓN DE TECNOLOGÍAS Y SERVICIOS INFORMÁTICOS

Universidad de Almería

Correo: jpj451@inlumine.ual.es

22 de octubre de 2025

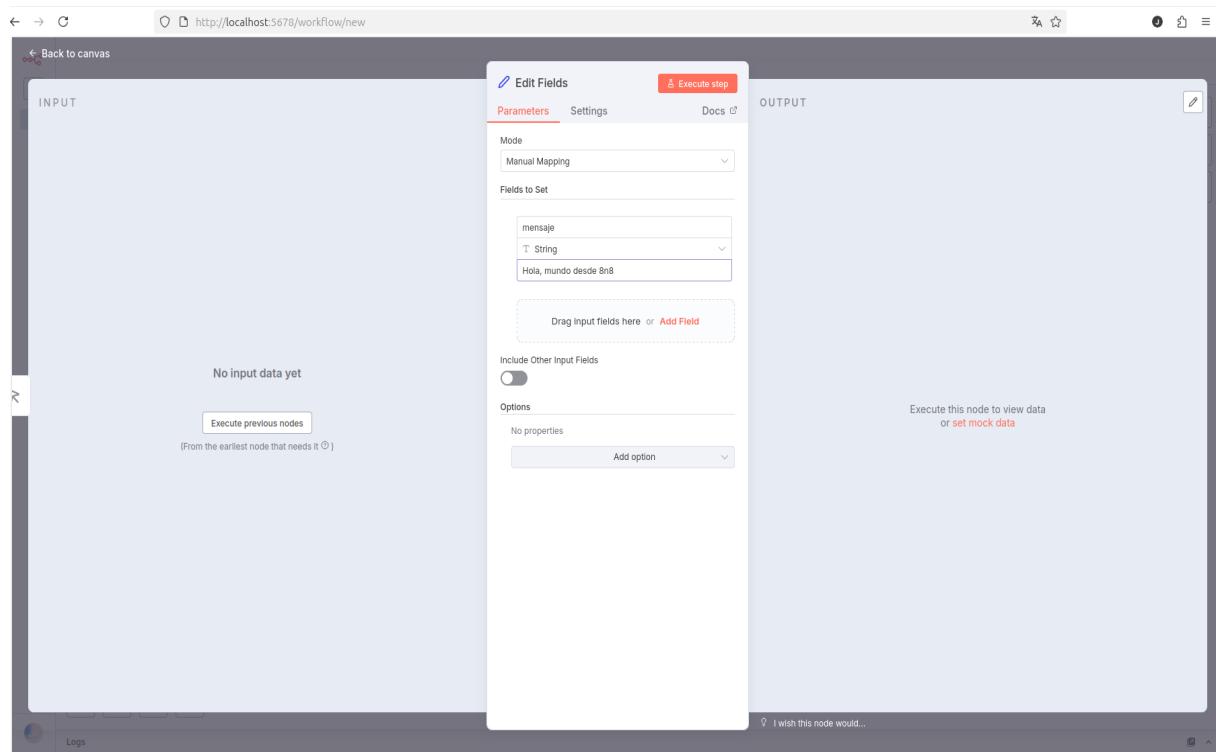
# Índice general

1. Ejercicio Guiado 1	2
2. Ejercicio Guiado 2	4
3. Ejercicio Guiado 3	6

# Capítulo 1

## Ejercicio Guiado 1

En el Ejercicio Guiado 1 el propósito será crear un flujo en el canvas para que muestre el mensaje “Hola, mundo desde n8n”. Para ello, lo primero que se debe hacer es añadir a la interfaz de n8n, la cual se le llama “Canvas”, un Manual Trigger que será el punto de partida ideal para el desarrollo y depuración del proyecto. A continuación se tiene que añadir el nodo Edits Fields (Set), útil para crear, modificar o eliminar campos de datos, donde se creará un valor de nombre “mensaje” y de valor “Hola, mundo desde n8n.”.

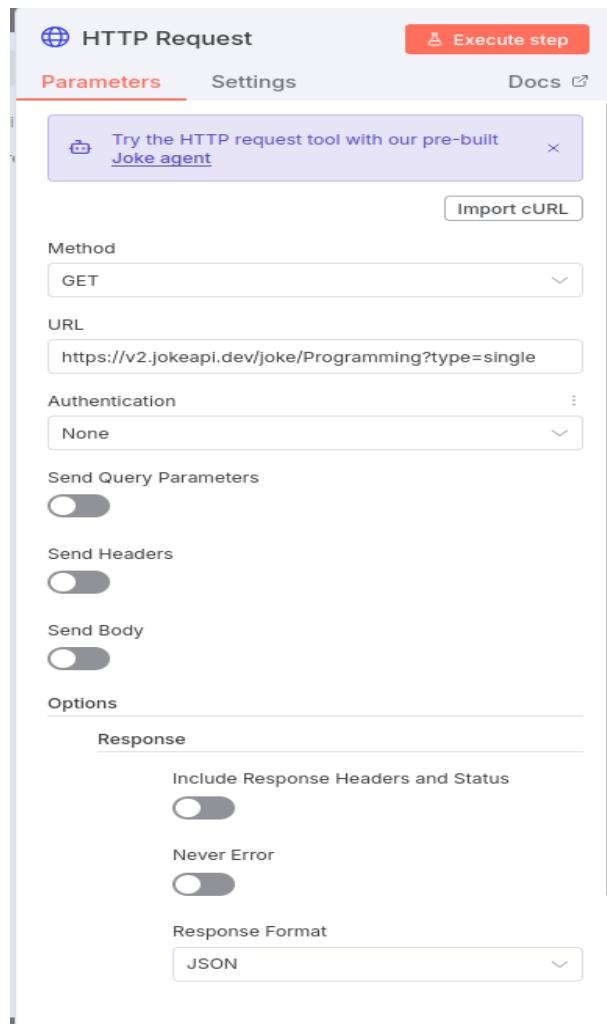


Una vez configurado el nodo correctamente se debe presionar el botón de “Test Step” el cual ejecutará el flujo hasta el nodo. Una vez ejecutado el nodo se podrá ver una vista “Table” donde se verá una columna “mensaje” y una fila con el valor “Hola,mundo desde n8n” y otra vista “JSON” para ver la estructura de datos subyacente donde se verá el array de objetos con el mensaje. A continuación se podrán ver dos fotos mostrando ambas vistas.

# Capítulo 2

## Ejercicio Guiado 2

El propósito del Ejercicio Guidao 2 será interactuar con una API REST Pública. Para ello se tendrá que introducir el nodo HTTP Request para consumir datos de un servicio externo. Una vez añadido habrá que configurarlo para poder consumir datos de una URL. En la configuración se debe definir el Método, la URL y el formato de respuesta.



Una vez se tenga el nodo configurado correctamente hay que ejecutarlo y mostrará un panel de resultados en la vista “JSON” con varios campos como error, category, type, joke, flogs, e id. De esta forma se podrá entender la estructura de los datos que se reciben.

The screenshot shows the MuleSoft Anypoint Studio interface with an open workflow canvas. A central node is an **HTTP Request** node. The **INPUT** tab shows a single item in a dropdown menu labeled "When clicking Execute workflow". The **OUTPUT** tab displays the received JSON response:

```
[{"error": false, "category": "Programming", "type": "single", "joke": "A man is smoking a cigarette and blowing smoke rings into the air. His girlfriend becomes irritated with the smoke and says \"Can't you see the warning on the cigarette pack? Smoking is hazardous to your health!\" to which the man replies, \"I am a programmer. We don't worry about warnings; we only worry about errors.\\"", "flags": {"nsfw": false, "religious": false, "political": false, "racist": false, "sexist": false, "explicit": false}, "id": 38, "safe": true, "lang": "en"}]
```

The interface includes tabs for **Schema**, **Table**, and **JSON**. At the bottom, there are buttons for **Logs**, **Add option**, and a note: "You can view the raw requests this node makes in your browser's developer console".

# Capítulo 3

## Ejercicio Guiado 3

En el Ejercicio Guiado 1 el propósito será consumir la respuesta “cruda” de la API y transformarla en un objeto de salida limpia. Para ello se debe añadir un node Edit Fields(Set) para añadirle expresiones y así limpiar las salidas. En la salida se quiere que solamente se muestre el texto, la categoría y la id. Para conseguir este propósito hay que añadir las expresiones “ json.joke ”, “ json.category ” y “ json.id ”.

The screenshot shows the MuleSoft Anypoint Studio interface with the 'Edit Fields1' node selected. The 'INPUT' tab displays an 'HTTP Request' item containing a single joke entry. The 'OUTPUT' tab shows the resulting JSON output. The 'Edit Fields1' configuration uses 'Manual Mapping' to transform the input fields into the desired output structure.

Mode	Fields to Set
Manual Mapping	broma = {{ \$json.joke }} A man is smoking a cigarette and blowing smoke rings into the air. His girlfriend becomes irritated with the smoke and says "Can't you see the warning on the cigarette pack? Smoking is hazardous to your health!" to which the man replies, "I am a programmer. We don't worry about warnings; we only worry about errors." String
Manual Mapping	categoria = {{ \$json.category }} Programming String
Manual Mapping	identificador = {{ \$json.id }} 38 String

The 'OUTPUT' tab shows the transformed JSON:

```
{
  "broma": "A man is smoking a cigarette and blowing smoke rings into the air. His girlfriend becomes irritated with the smoke and says 'Can't you see the warning on the cigarette pack? Smoking is hazardous to your health!' to which the man replies, 'I am a programmer. We don't worry about warnings; we only worry about errors.'",
  "categoria": "Programming",
  "identificador": "38"
}
```