

Universidad de Almería

Máster en Ingeniería Informática

Integración de Tecnologías y Servicios Informáticos

Práctica 1

Introducción a n8n

Autor: Johan Eduardo Cala Torra

Fecha: 8 de diciembre de 2025

Curso 2024-2025

Índice

1. Introducción	2
2. Configuración del Entorno	2
2.1. Requisitos Previos	2
2.2. Despliegue de n8n con Docker	2
2.3. Acceso a n8n	3
3. Ejercicio 1: Hola Mundo	4
3.1. Paso 1: Añadir Manual Trigger	4
3.2. Paso 2: Configurar Edit Fields (Set)	4
3.3. Paso 3: Ejecución y Resultado	5
4. Ejercicio 2: Consumo de API REST	5
4.1. Paso 1: Configurar HTTP Request	5
4.2. Paso 2: Análisis de la Respuesta	6
5. Ejercicio 3: Transformación de Datos	6
5.1. Paso 1: Mapeo de Campos	6
5.2. Paso 2: Resultado Final	8
6. Flujo de Trabajo Completo	8
7. Anexo: Archivo JSON del Workflow	8

1. Introducción

Esta práctica tiene como objetivo introducir los conceptos fundamentales de **n8n**, una plataforma de automatización de flujos de trabajo de tipo *low-code*. A lo largo de la práctica se han cubierto los siguientes aspectos:

- Despliegue de n8n utilizando Docker
- Comprensión de la arquitectura basada en nodos
- Creación de flujos de trabajo básicos
- Consumo de APIs REST externas
- Transformación de datos

2. Configuración del Entorno

2.1. Requisitos Previos

Para realizar esta práctica se requiere tener instalado:

- Docker Desktop (Windows/macOS) o Docker Engine (Linux)
- Un navegador web moderno

2.2. Despliegue de n8n con Docker

Se utilizó Docker Compose para desplegar n8n de forma persistente. El archivo `docker-compose.yml` contiene la configuración necesaria.

```

PS C:\Users\JohanCalaMaster\OneDrive\Escritorio\PracticalN8n> docker run -it
--rm --name n8n -p 5678:5678 -v ${PWD}/.n8n:/home/node/.n8n n8nio/n8n
Unable to find image 'n8nio/n8n:latest' locally
latest: Pulling from n8nio/n8n
3cdbfff583b0: Pull complete
a15eca46077e: Pull complete
f479db9059a2: Pull complete
d17558741c28: Pull complete
4f4fb700ef54: Pull complete
435cf53991d4: Pull complete
9824c27679d3: Pull complete
d98c9de455e9: Pull complete
2d381af09514: Pull complete
5e8dc67b0f45: Pull complete
e62ad5808fa8: Pull complete
0939f1730a4c: Pull complete
Digest: sha256:14248a2a18487c3e1669d2e2bf797f0648f606e0c168e36ef20c3a1f47
eac
Status: Downloaded newer image for n8nio/n8n:latest
No encryption key found - Auto-generating and saving to: /home/node/.n8n/con
fig
Permissions 0644 for n8n settings file /home/node/.n8n/config are too wide.
This is ignored for now, but in the future n8n will attempt to change the pe
rmissions automatically. To automatically enforce correct permissions now se
t N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true (recommended), or turn this che
ck off set N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=false.
Initializing n8n process
n8n ready on ::, port 5678
Migrations in progress, please do NOT stop the process.
Starting migration InitialMigration1588102412422
Finished migration InitialMigration1588102412422
Starting migration WebhookModel1592445003908
Finished migration WebhookModel1592445003908
Starting migration CreateIndexStoppedAt1594825041918
Finished migration CreateIndexStoppedAt1594825041918
Starting migration MakeStoppedAtNullable1607431743769
Finished migration MakeStoppedAtNullable1607431743769
Starting migration AddWebhookId1611071044839

```

Figura 1: Ejecución del comando para iniciar n8n

2.3. Acceso a n8n

Una vez iniciado el contenedor, se accede a n8n a través del navegador en la dirección <http://localhost:5678>.

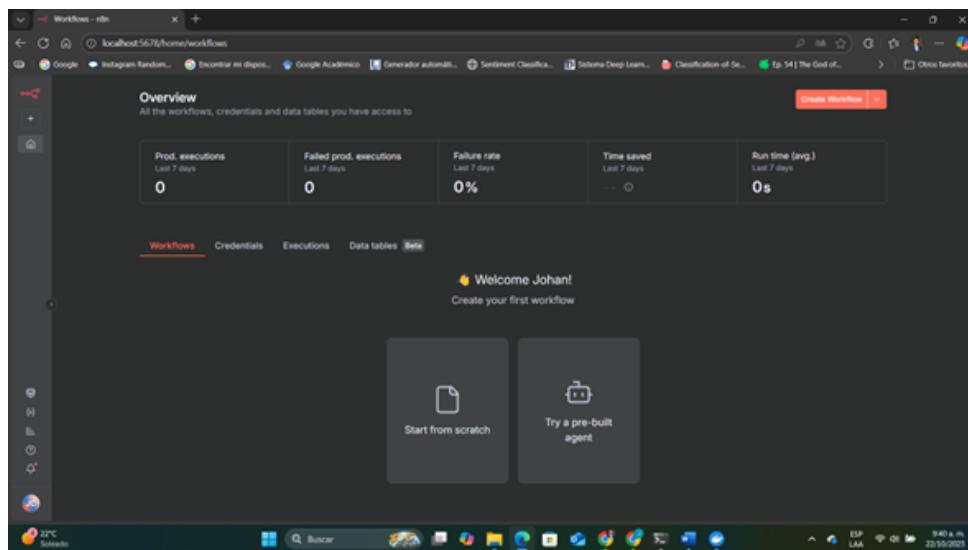


Figura 2: Configuración inicial de n8n

3. Ejercicio 1: Hola Mundo

El primer ejercicio consiste en crear un flujo de trabajo básico que demuestre la estructura de datos de n8n.

3.1. Paso 1: Añadir Manual Trigger

Se inicia creando un nuevo workflow y añadiendo el nodo **Manual Trigger**, que permite ejecutar el flujo de forma manual.

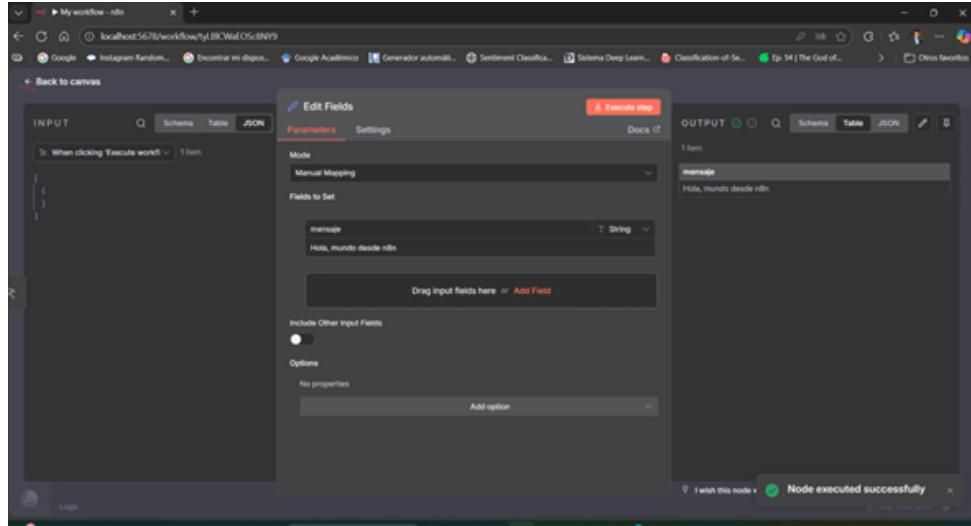


Figura 3: Añadir nodo Manual Trigger

3.2. Paso 2: Configurar Edit Fields (Set)

Se añade el nodo **Edit Fields (Set)** y se configura con un campo llamado **mensaje** con el valor "Hola, mundo desde n8n".

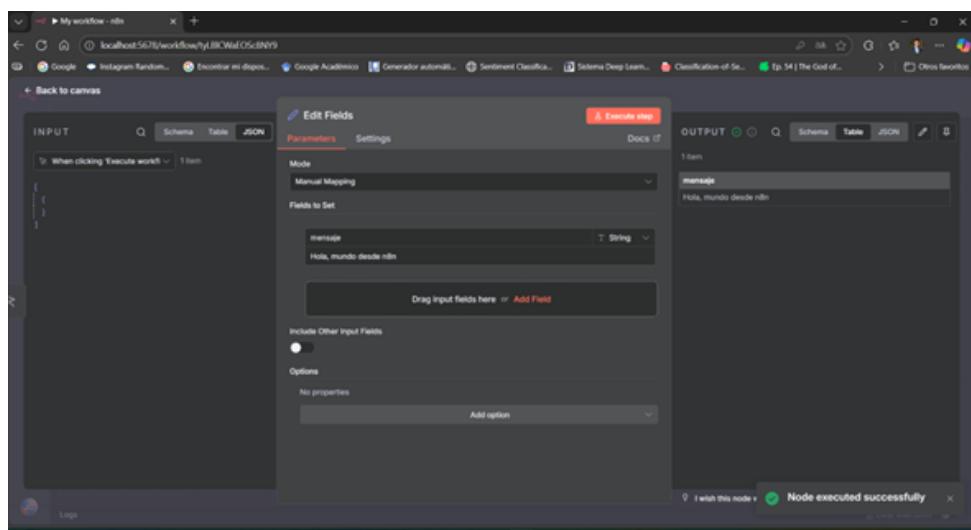
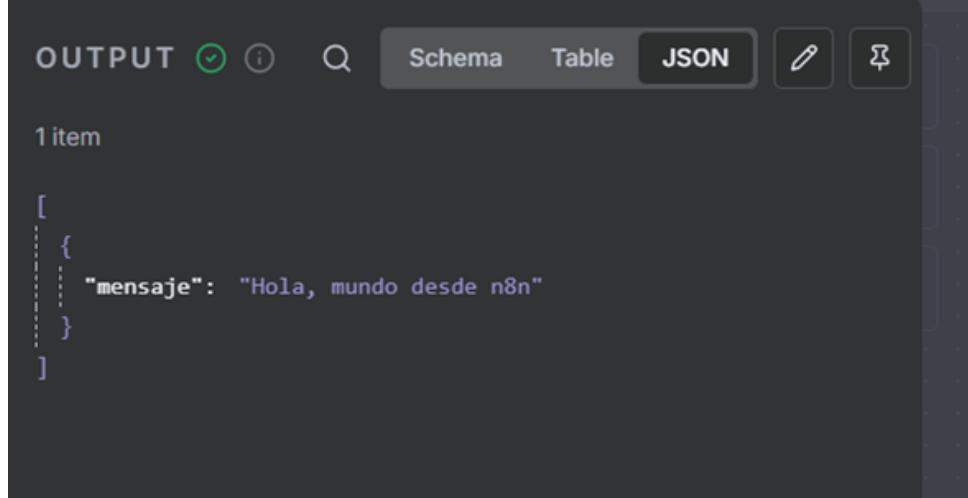


Figura 4: Configuración del nodo Edit Fields (Set)

3.3. Paso 3: Ejecución y Resultado

Al ejecutar el flujo, se observa la estructura de datos de n8n: un array de objetos con la clave json.



```
[{"mensaje": "Hola, mundo desde n8n"}]
```

Figura 5: Resultado del Ejercicio 1

4. Ejercicio 2: Consumo de API REST

En este ejercicio se utiliza el nodo **HTTP Request** para consumir datos de una API pública (JokeAPI).

4.1. Paso 1: Configurar HTTP Request

Se añade el nodo HTTP Request con la siguiente configuración:

- **Method:** GET
- **URL:** <https://v2.jokeapi.dev/joke/Programming?type=single>

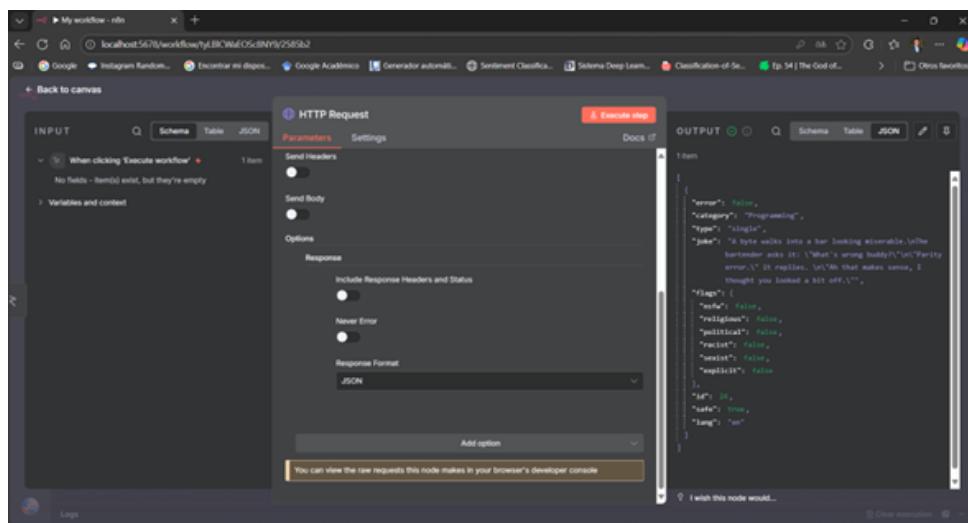


Figura 6: Configuración del nodo HTTP Request

4.2. Paso 2: Análisis de la Respuesta

La API devuelve un objeto JSON con campos como `joke`, `category`, `id`, entre otros.

Ejercicio 2 - HTTP Request Success in 186ms							
OUTPUT				Input	Output	...	
error	category	type	joke	flags	id	safe	lang
false	Programming	single	UDP is better in the COVID era since it avoids unnecessary handshakes.	<code>nsfw : false</code> <code>religious : false</code> <code>political : false</code> <code>racist : false</code> <code>sexist : false</code> <code>explicit : false</code>	258	true	en

Figura 7: Respuesta de la API JokeAPI

5. Ejercicio 3: Transformación de Datos

El objetivo es transformar la respuesta “cruda” de la API en un formato limpio con solo los campos necesarios.

5.1. Paso 1: Mapeo de Campos

Se añade otro nodo **Edit Fields (Set)** con las siguientes expresiones:

- `broma: {{ $json.joke }}`
- `categoria: {{ $json.category }}`
- `identificador: {{ $json.id }}`

Edit Fields Execute step

Parameters Settings Docs

broma

Type: String

= `{{ $json.joke }}`

A byte walks into a bar looking m...

categoria

Type: String

= `={{ $json.category }}`

=Programming

identificador

Type: String

= `{{ $json.id }}`

Result Item 0 < >

24

Tip: Type . for data transformation options. [Learn more](#)

Include Other Input Fields

Options

Figura 8: Mapeo de campos con expresiones

5.2. Paso 2: Resultado Final

El flujo completo transforma los datos de la API a un formato limpio.

The screenshot shows the 'Edit Fields' step in the n8n workflow editor. The input node is an 'HTTP Request' node. The transformation step is titled 'Edit Fields' with the following parameters:

- broma: {{ \$item.joke }}
- category: {{ \$item.category }}
- identificador: {{ \$item.id }}

The output node shows a single item with the following structure:

broma	categoria	Identificador
A byte walks into a bar looking miserable... The bartender asks it: "What's wrong?" It replies, "\A that makes sense, I thought you looked a bit off."	Programming	24

Figura 9: Resultado del Ejercicio 3 - Datos transformados

6. Flujo de Trabajo Completo

La Figura 10 muestra el flujo de trabajo completo con todos los nodos conectados.

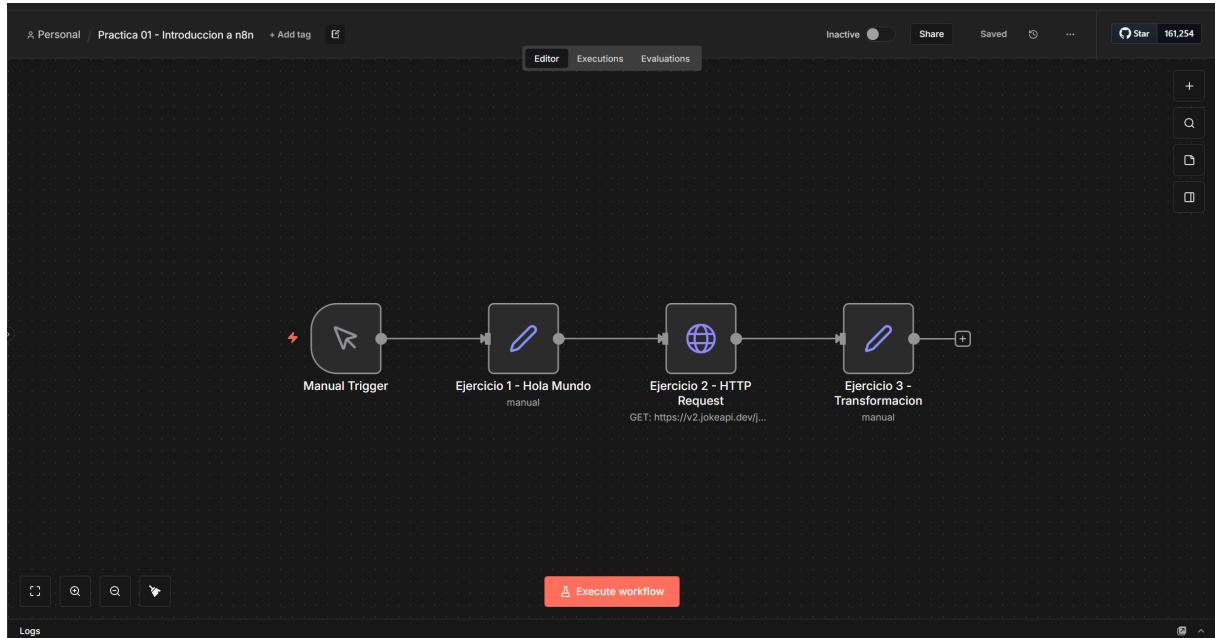


Figura 10: Flujo de trabajo completo de la Práctica 1

7. Anexo: Archivo JSON del Workflow

El archivo JSON del flujo de trabajo se encuentra en: `workflows/practica-01-workflow.json`. Este archivo puede importarse en n8n desde el menú **Workflows → Import from File**.