

**Universidad de Almería**

Máster en Ingeniería Informática

## **Integración de Tecnologías y Servicios Informáticos**

---

### **Práctica 2**

Lógica, Programación y Manejo de Colecciones en n8n

---

**Autor:** Johan Eduardo Cala Torra

**Fecha:** 8 de diciembre de 2025

Curso 2024-2025

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Ejercicio 1: Filtrado por Categoría</b>	<b>2</b>
2.1. Objetivo . . . . .	2
2.2. Configuración del Schedule Trigger . . . . .	2
2.3. HTTP Request con Múltiples Categorías . . . . .	2
2.4. Split Out para Separar Chistes . . . . .	3
2.5. Nodo IF para Clasificar por Categoría . . . . .	3
2.6. Resultado del Ejercicio 1 . . . . .	4
<b>3. Ejercicio 2: Análisis de Productos</b>	<b>5</b>
3.1. Objetivo . . . . .	5
3.2. Schedule Trigger Diario . . . . .	5
3.3. HTTP Request a FakeStoreAPI . . . . .	5
3.4. Procesamiento Automático de Arrays en n8n . . . . .	6
3.5. Nodo IF por Precio . . . . .	6
3.6. Resultado del Ejercicio 2 . . . . .	7
<b>4. Ejercicio 3: Planificador de Festivos</b>	<b>7</b>
4.1. Objetivo . . . . .	7
4.2. HTTP Request - Festivos de España . . . . .	8
4.3. Filtrado por Festivo Nacional . . . . .	8
4.4. HTTP Request - Bored API con Batching . . . . .	8
4.5. Combinación de Datos . . . . .	9
4.6. Resultado del Ejercicio 3 . . . . .	10
<b>5. Conclusiones</b>	<b>10</b>
5.1. Lecciones Aprendidas . . . . .	10

# 1. Introducción

Esta práctica profundiza en las capacidades de n8n, transitando desde flujos manuales simples hacia automatizaciones robustas con lógica de negocio. Se abordan los siguientes conceptos:

- **Schedule Trigger:** Ejecución automática y programada de workflows
- **Nodo IF:** Implementación de lógica condicional y bifurcaciones
- **Split Out:** Procesamiento de colecciones y arrays de datos

## 2. Ejercicio 1: Filtrado por Categoría

### 2.1. Objetivo

Modificar el flujo de trabajo guiado para clasificar chistes según su categoría: “Programming” o “Misc”.

### 2.2. Configuración del Schedule Trigger

Se configura el disparador para ejecutarse cada minuto durante las pruebas.

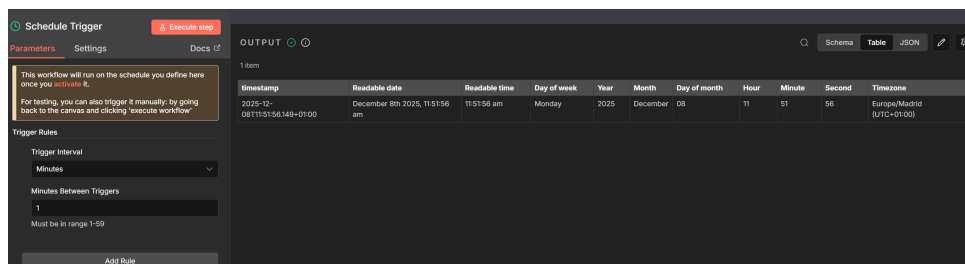


Figura 1: Configuración del Schedule Trigger

### 2.3. HTTP Request con Múltiples Categorías

Se modifica la URL para solicitar chistes de ambas categorías: `Programming,Misc`.

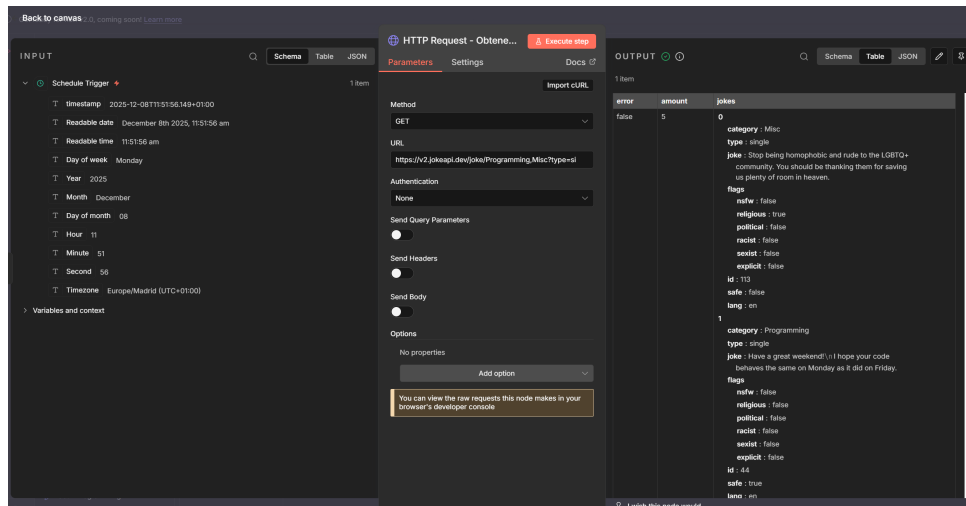


Figura 2: Configuración del HTTP Request con múltiples categorías

## 2.4. Split Out para Separar Chistes

El nodo Split Out descompone el array jokes en ítems individuales.

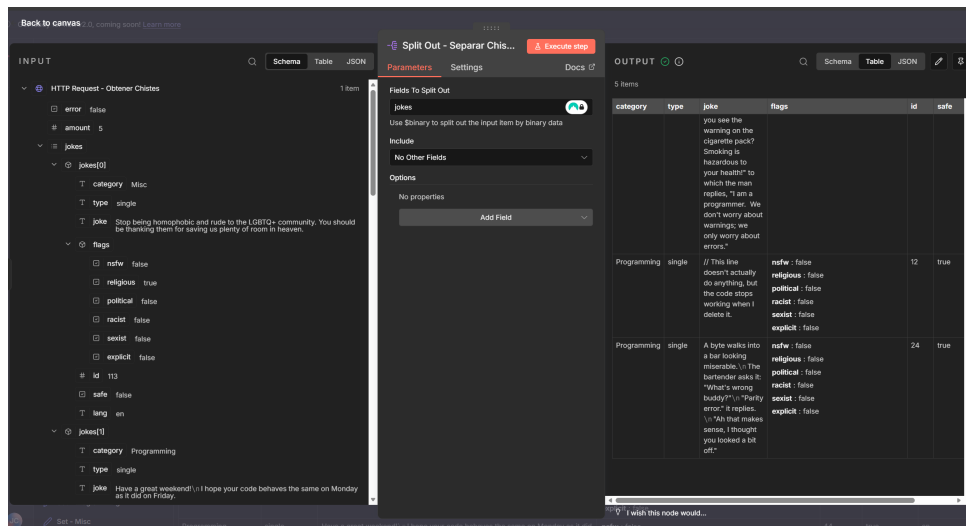


Figura 3: Configuración del nodo Split Out

## 2.5. Nodo IF para Clasificar por Categoría

Se evalúa si el campo category es igual a "Programming".

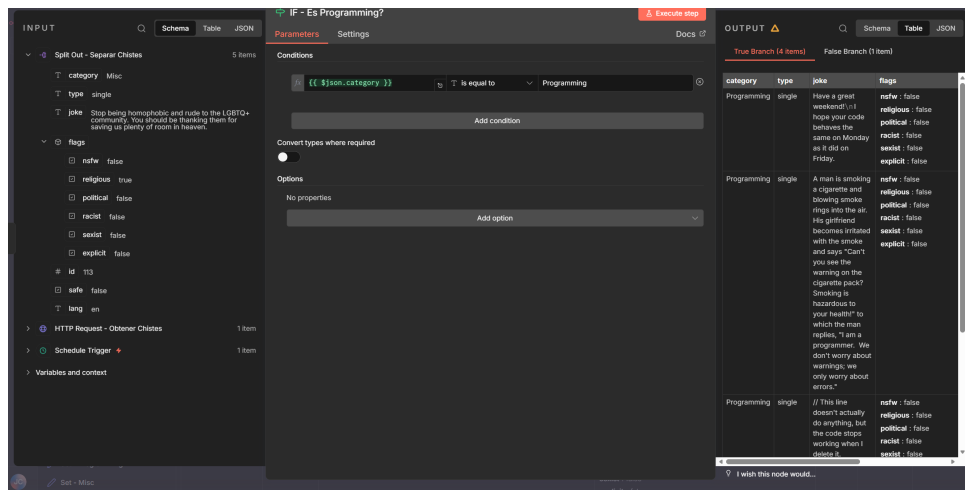


Figura 4: Configuración del nodo IF

## 2.6. Resultado del Ejercicio 1

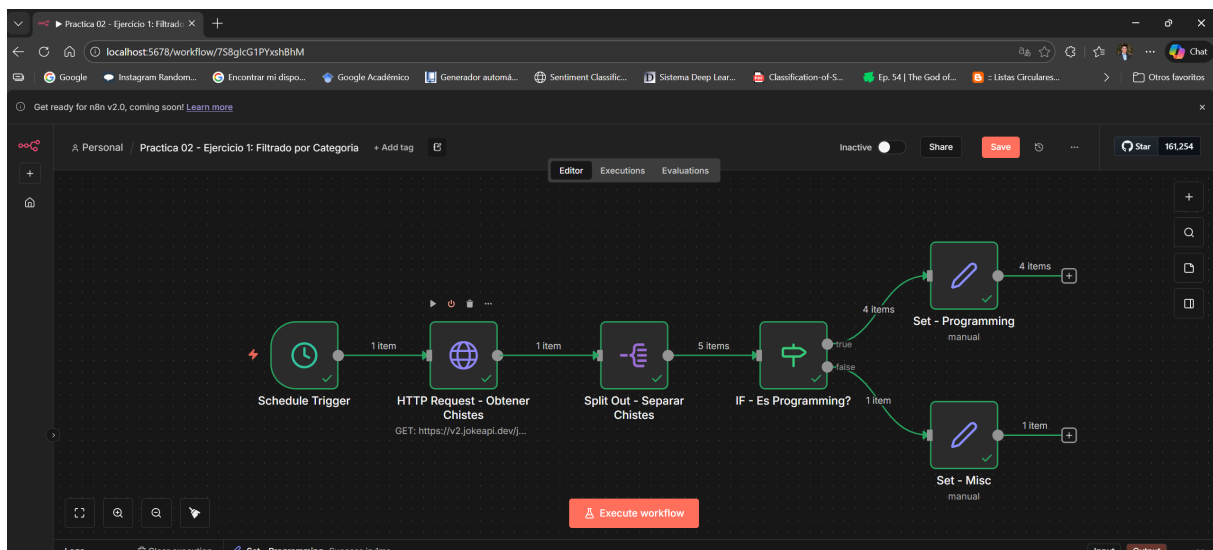


Figura 5: Workflow completo del Ejercicio 1

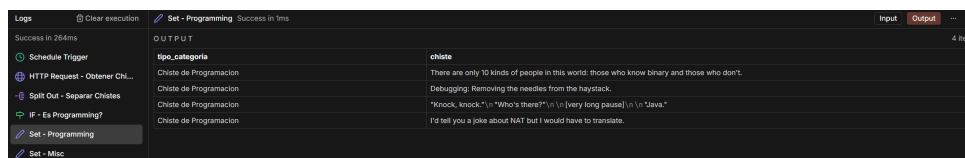


Figura 6: Resultado de la ejecución - Chistes clasificados

## 3. Ejercicio 2: Análisis de Productos

### 3.1. Objetivo

Crear un flujo que analice productos de FakeStoreAPI, clasificándolos como “Premium” (precio  $> 100$ ) o “Estándar” (precio  $\leq 100$ ).

### 3.2. Schedule Trigger Diario

Se configura el disparador para ejecutarse una vez al día.

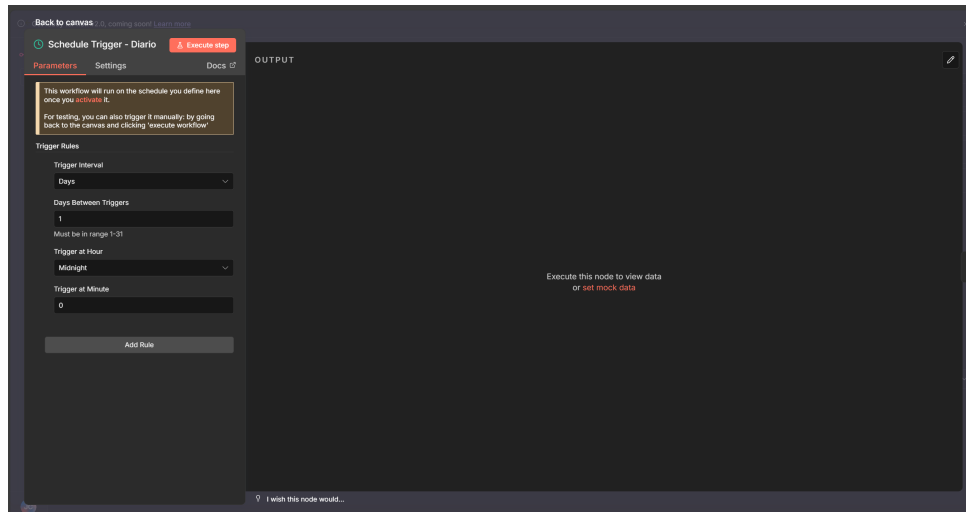


Figura 7: Schedule Trigger configurado para ejecución diaria

### 3.3. HTTP Request a FakeStoreAPI

Se obtiene la lista completa de productos desde <https://fakestoreapi.com/products>.

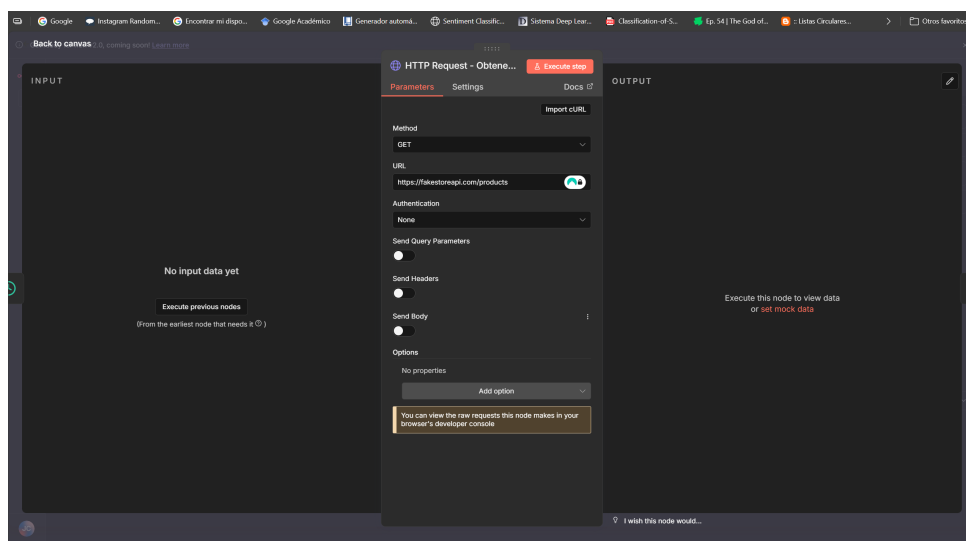


Figura 8: HTTP Request a FakeStoreAPI

### 3.4. Procesamiento Automático de Arrays en n8n

**Nota importante:** Durante la implementación se descubrió que n8n procesa arrays automáticamente. Según la documentación oficial: *“n8n nodes take any number of items as input, process these items, and output the results... Nodes usually run once for each item.”*

Esto significa que cuando el HTTP Request devuelve un array de 20 productos, el nodo IF se ejecuta automáticamente 20 veces (una por cada producto), eliminando la necesidad de usar Split Out cuando la API devuelve un array directamente en la raíz de la respuesta.

### 3.5. Nodo IF por Precio

Se evalúa si el precio es mayor a 100 para clasificar cada producto.

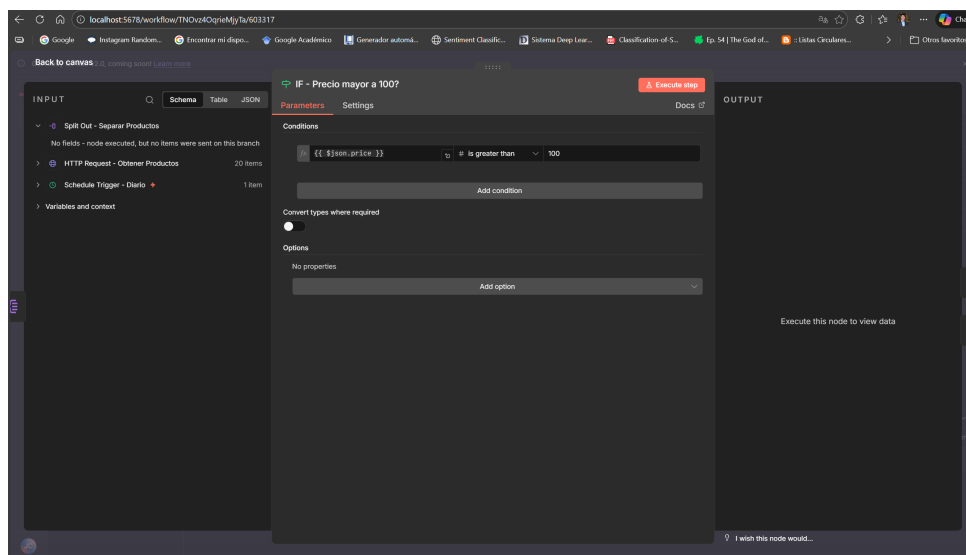


Figura 9: Configuración del IF para evaluar precio ¿100

## 3.6. Resultado del Ejercicio 2

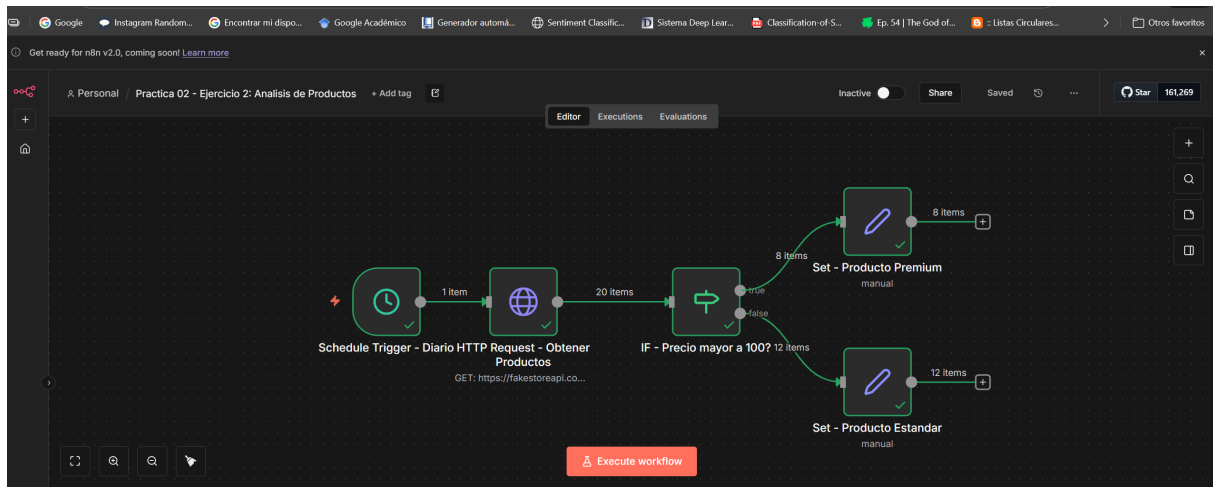


Figura 10: Workflow completo del Ejercicio 2

The screenshot shows the execution logs for the "Set - Producto Premium" node, which successfully executed in 3ms. The output is a table of 8 items:

id	title	price	description	category	image	rating
1	Fallraven - Foldback No. 1 Backpack, Fits 15 Laptops	109.99	Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday	men's clothing	https://fakestoreapi.com/img/8FkPd-2AVl_AC_SL1500_1.png	rate: 3.9 count: 120
5	John Hardy Women's Legends Naga Gold & Silver Dragon Station Chain Bracelet	695	From our Legends Collection, the Naga was inspired by the mythical water dragon that protects the ocean's pearl. Wear facing inward to be bestowed with love and abundance, or outward for protection.	jewelry	https://fakestoreapi.com/img/71pWzhdnNwL_AC_UL640_QL65_ML3_1.png	rate: 4.6 count: 400
6	Solid Gold Petite Micropave	168	Satisfaction Guaranteed. Return or exchange any order within 30 days. Designed and sold by Hafeez Center in the United States. Satisfaction Guaranteed. Return or exchange any order within 30 days.	jewelry	https://fakestoreapi.com/img/61bmMuC6uL_AC_UL640_QL65_ML3_1.png	rate: 3.9 count: 70
10	SanDisk SSD PLUS 1TB Internal SSD - SATA III 6 Gb/s	109	Easy upgrade for faster boot up, shutdown, application load and response (As compared to 5400 RPM SATA 2.5" hard drives. Based on published specifications and internal	electronics	https://fakestoreapi.com/img/61uTtKoQl_AC_SX679_1.png	rate: 2.9 count: 470

Figura 11: Productos clasificados como Premium

The screenshot shows the execution logs for the "Set - Producto Estandar" node, which successfully executed in 4ms. The output is a table of 12 items (showing the first 4):

id	title	price	description	category	image	rating
2	Mens Casual Premium Slim Fit T-Shirts	22.3	Slim-fitting style, contrast raglan long sleeve, three-button henley placket, light weight & soft fabric for breathable and comfortable wearing. And Solid stitched shirts with round neck made for durability and a great fit for casual fashion wear and dihard baseball fans. The Henley style round neckline includes a three-button placket.	men's clothing	https://fakestoreapi.com/img/71qndwUjXAC_SL1500_1.png	rate: 4.1 count: 259
3	Mens Cotton Jacket	55.99	great outerwear jackets for Spring/Autumn/Winter, suitable for many occasions, such as working, hiking, camping, mountain/rock climbing, cycling, traveling or other outdoors. Good gift choice for you or your family member. A warm hearted love to Father, husband or son in this Thanksgiving or Christmas Day.	men's clothing	https://fakestoreapi.com/img/71u3QwUjXAC_SL1500_1.png	rate: 4.7 count: 500
4	Mens Casual Slim Fit	15.99	The color could be slightly different between on	men's clothing	https://fakestoreapi.com/img/71YXzeOusL_AC_U	rate: 2.1

Figura 12: Productos clasificados como Estándar

## 4. Ejercicio 3: Planificador de Festivos

### 4.1. Objetivo

Obtener festivos nacionales de España 2025 y asignar una actividad aleatoria a cada uno usando la Bored API.



## 4.2. HTTP Request - Festivales de España

Se consulta la API de Nager.Date para obtener los festivales de 2025.

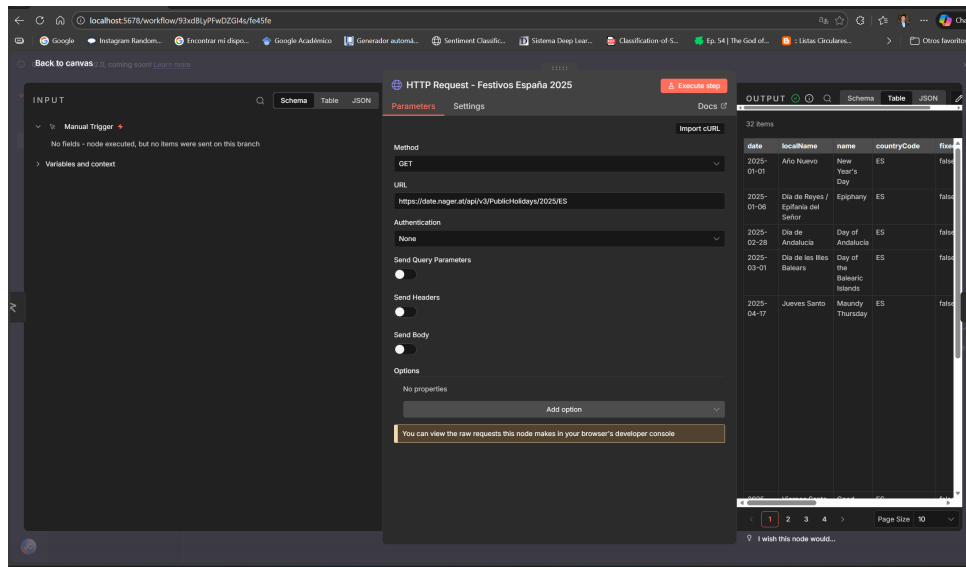


Figura 13: HTTP Request a la API de festivales

## 4.3. Filtrado por Festivo Nacional

Al igual que en el Ejercicio 2, n8n procesa automáticamente el array de festivales. Se evalúa el campo `global` para identificar festivales nacionales.

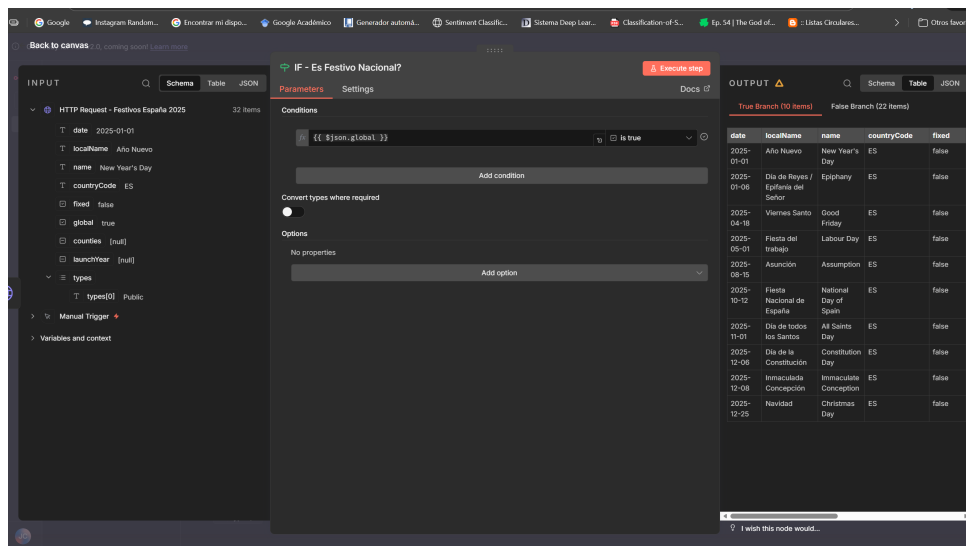


Figura 14: IF para filtrar festivales nacionales (`global = true`)

## 4.4. HTTP Request - Bored API con Batching

Para cada festivo nacional, se obtiene una actividad aleatoria.

**Problema encontrado:** Al ejecutar el workflow, la Bored API devolvía el error *“The service is receiving too many requests from you”* debido a que n8n por defecto procesa las

peticiones en paralelo (las 10 peticiones simultáneamente).

**Solución:** Se implementó **batching** en el nodo HTTP Request con las siguientes configuraciones:

- **batchSize:** 1 - Procesar una petición a la vez
- **batchInterval:** 1000 - Esperar 1 segundo entre cada petición

Esto permite controlar el rate limiting de APIs externas, pasando de procesamiento paralelo (rápido pero problemático) a procesamiento secuencial (controlado).

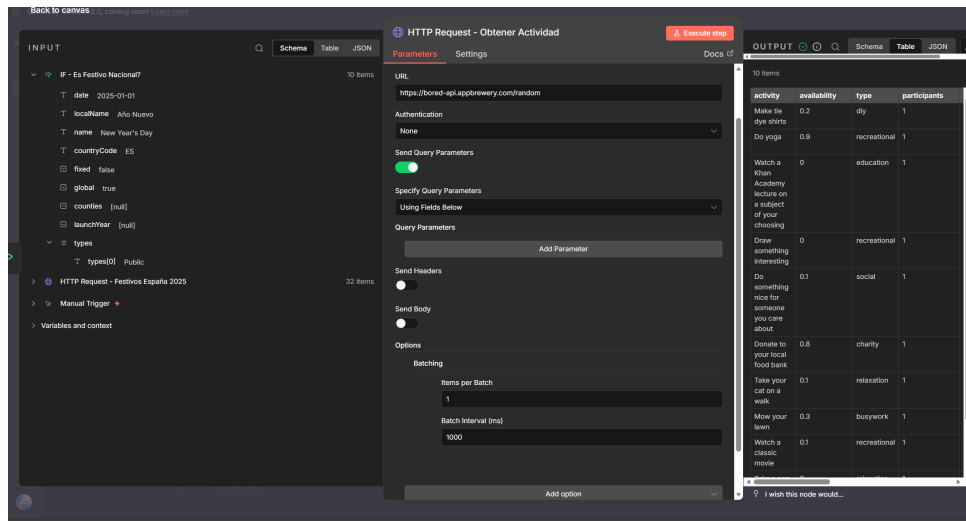


Figura 15: HTTP Request a la Bored API con configuración de batching

## 4.5. Combinación de Datos

Se utiliza Edit Fields para combinar la información del festivo con la actividad.

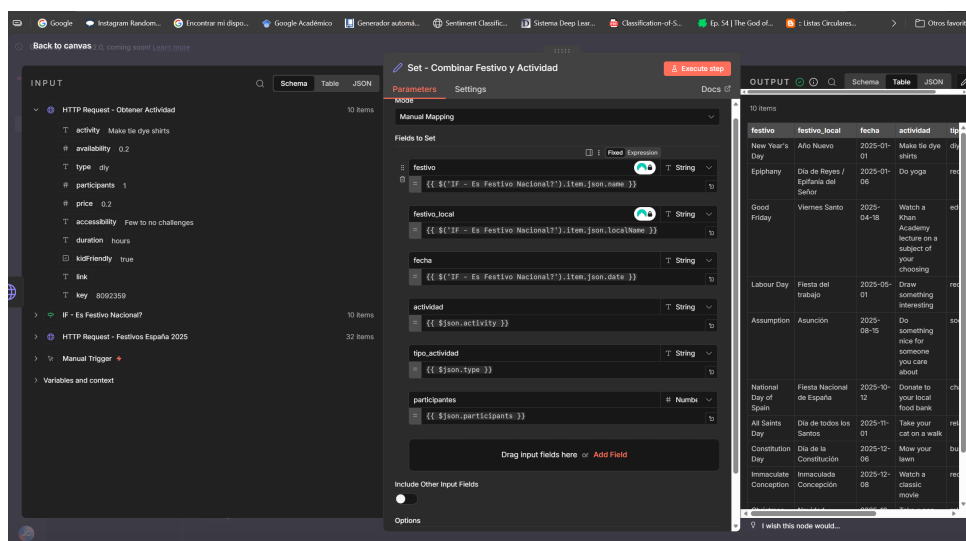


Figura 16: Configuración para combinar festivo y actividad

## 4.6. Resultado del Ejercicio 3

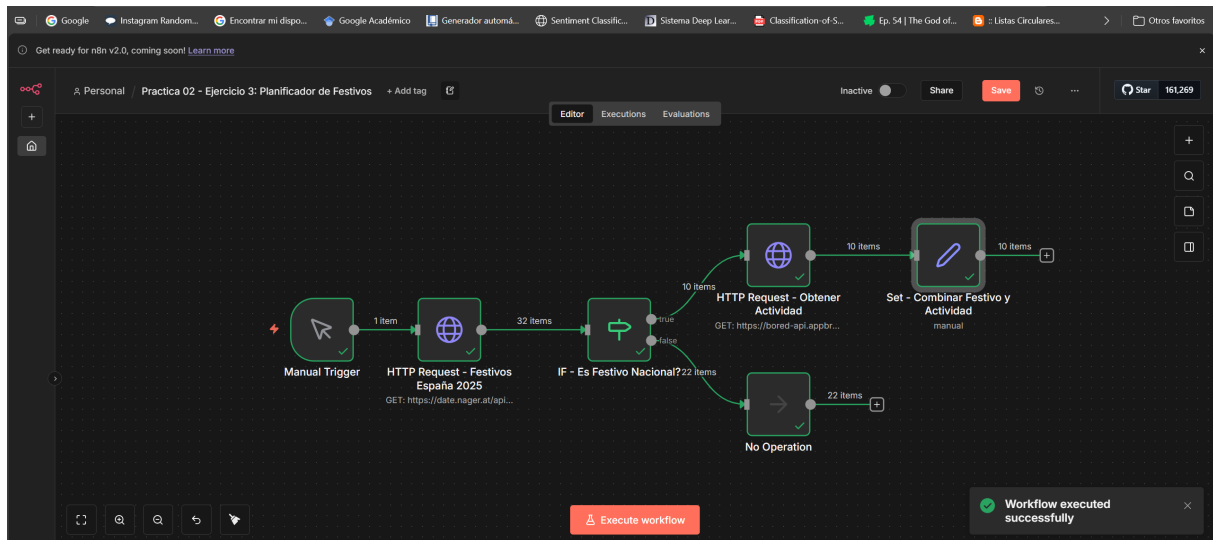


Figura 17: Workflow completo del Ejercicio 3

activity	availability	type	participants	price	accessibility	duration	kidfriendly	link	key
Plant a tree	0.3	recreational	1	0.3	Minor challenges	hours	true	empty	1942393
Do something you used to do as a kid	0.8	relaxation	1	0.1	Few to no challenges	minutes	true	empty	8827573
Go for a walk	0.1	relaxation	1	0	Few to no challenges	minutes	true	empty	4286250
Volunteer and help out at a senior center	0.1	charity	1	0	Few to no challenges	hours	true	empty	3920096
Play a video game	0	recreational	1	0	Few to no challenges	hours	true	empty	5534113
Binge watch a trending series	0.2	recreational	1	0.2	Few to no challenges	hours	true	empty	5884647
Start a webinar on a topic of your choice	0.9	education	1	0	Few to no challenges	hours	true	empty	6826029
Text a friend you haven't talked to in a long time	0.2	social	2	0.05	Few to no challenges	minutes	true	empty	6087071
Hold a yard sale	0.1	social	1	0	Minor challenges	days	true	empty	1432113
Find a DIY to do	0.3	diy	1	0.4	Few to no challenges	minutes	true	empty	4981818

Figura 18: Resultado final: Festivales nacionales con actividades

## 5. Conclusiones

En esta práctica se han alcanzado los siguientes objetivos:

1. Configuración de **Schedule Trigger** para automatización temporal
2. Uso del nodo **IF** para implementar lógica condicional
3. Comprensión del nodo **Split Out** para procesar colecciones de datos
4. Consumo de múltiples APIs REST en un mismo workflow
5. Combinación de datos de diferentes fuentes

### 5.1. Lecciones Aprendidas

Durante la implementación se descubrieron aspectos importantes del funcionamiento interno de n8n:

- **Procesamiento automático de arrays:** n8n procesa arrays automáticamente sin necesidad de Split Out cuando la API devuelve un array directamente en la raíz de la respuesta.
- **Batching para rate limiting:** Cuando una API tiene límites de peticiones (rate limits), se debe usar la opción de batching (`batchSize` y `batchInterval`) para procesar secuencialmente con pausas en lugar del procesamiento paralelo por defecto.
- **Procesamiento paralelo vs secuencial:** Por defecto n8n ejecuta las peticiones en paralelo (rápido), pero esto puede causar problemas con APIs que tienen límites. El batching permite cambiar a procesamiento secuencial (controlado).

Estos conceptos son fundamentales para crear automatizaciones robustas en entornos de integración de sistemas.