

Práctica 2: Lógica, Programación y Manejo de Colecciones en n8n

Realizado por: Anass El Jabiry Kaddir

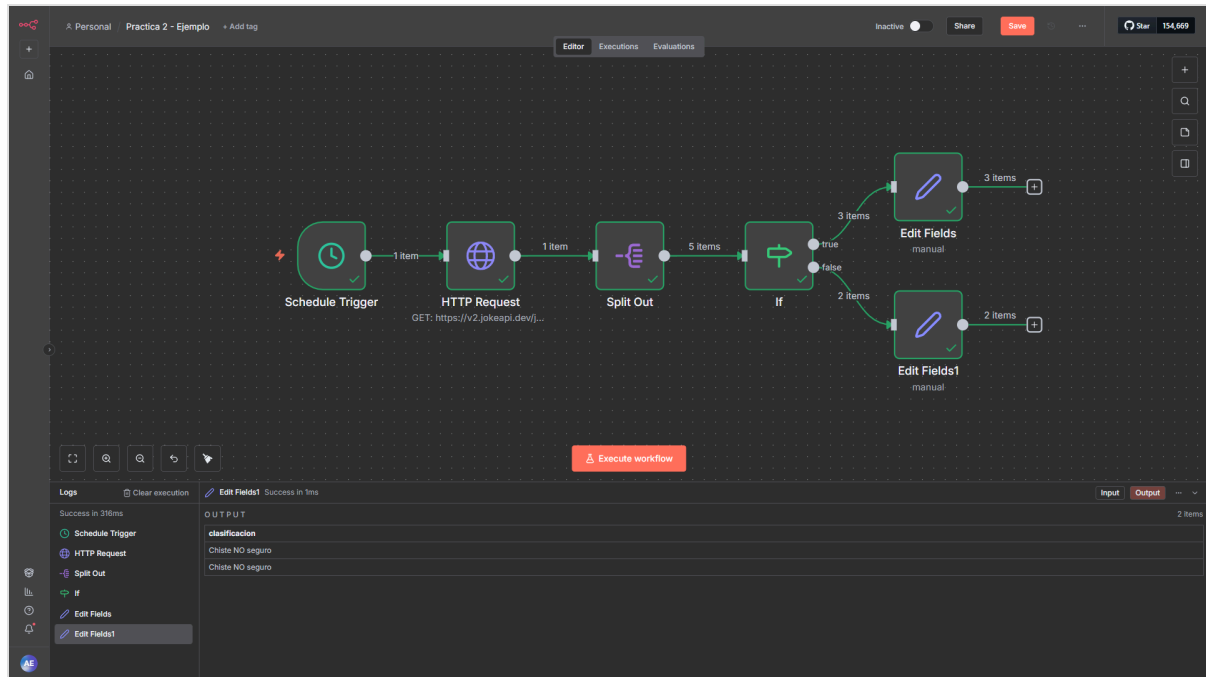
Desarrollo del Flujo de Trabajo Guiado

En este flujo guiado, hemos construido un proceso automatizado que se ejecuta periódicamente. El flujo comienza con un `Schedule Trigger` (disparador programado) que, para la prueba, se configuró para activarse cada minuto.

A continuación, un nodo `HTTP Request` consulta la API de chistes (JokeAPI) para obtener una colección de 5 chistes de programación. Como la API devuelve los 5 chistes en un solo array, utilizamos el nodo `Split Out` para separar esta colección en 5 ítems individuales.

Posteriormente, un nodo `IF` evalúa cada chiste de forma independiente comprobando la propiedad booleana `safe`. Finalmente, el flujo se bifurca: si `safe` es `true`, un nodo `Edit Fields (Set)` añade el campo `clasificacion` con el valor 'Chiste seguro'. Si es `false`, otro nodo `Edit Fields (Set)` añade el valor 'Chiste NO seguro'. La captura de pantalla muestra el flujo completo y su

lógica condicional.

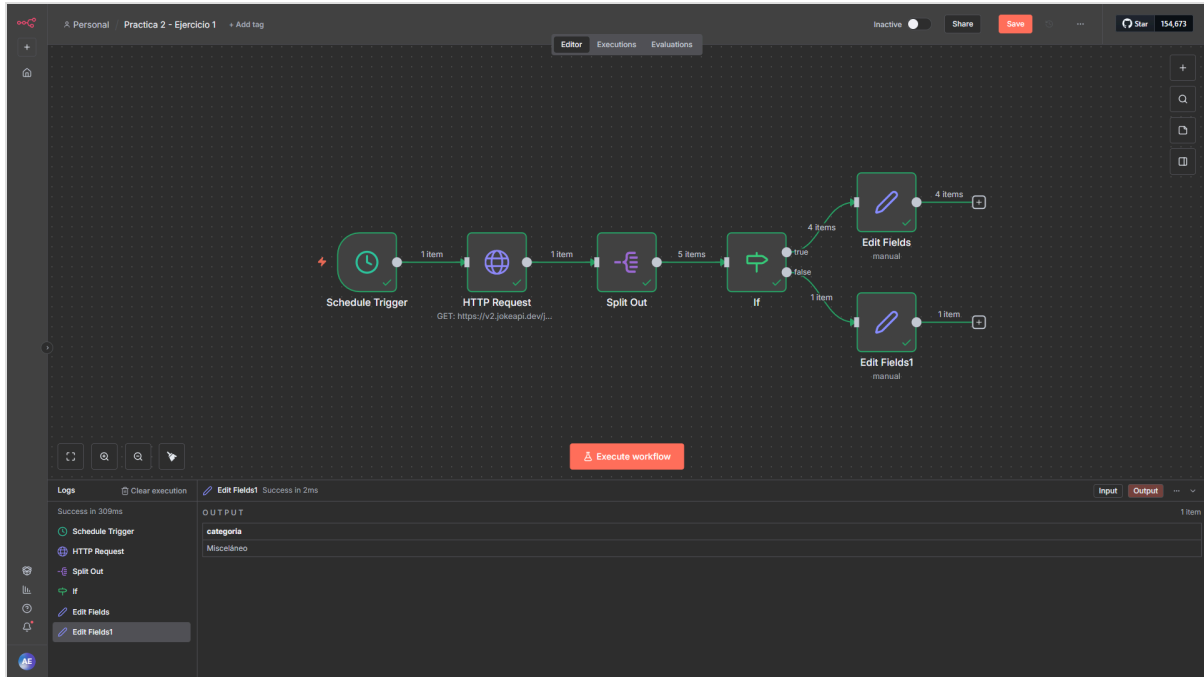


Ejercicio 1: Filtrado por Categoría (Dificultad: Baja)

Para este ejercicio, se modificó el flujo de trabajo guiado. El primer cambio fue en el nodo `HTTP Request`, donde se ajustó la URL para solicitar chistes de las categorías "Programming" y "Misc" simultáneamente, tal como permite la documentación de JokeAPI.

Después del nodo `split out` (que se mantiene para procesar cada chiste), se reconfiguró el nodo `if`. En lugar de comprobar el campo `safe`, la condición se ajustó para evaluar si el campo `category` es igual a "Programming".

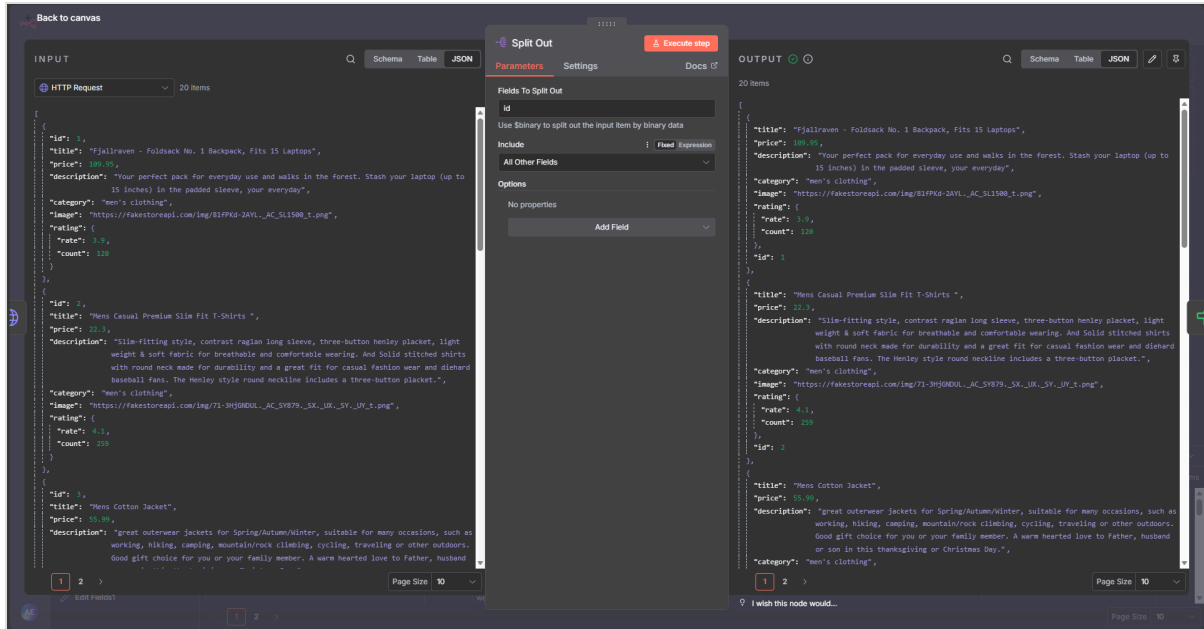
Finalmente, los nodos `Edit Fields` (Set) en las ramas `true` y `false` se modificaron para añadir un nuevo campo (ej. `clasificacion_cat`) que refleja si el chiste pertenece a 'Programming' o a 'Misc', respectivamente.



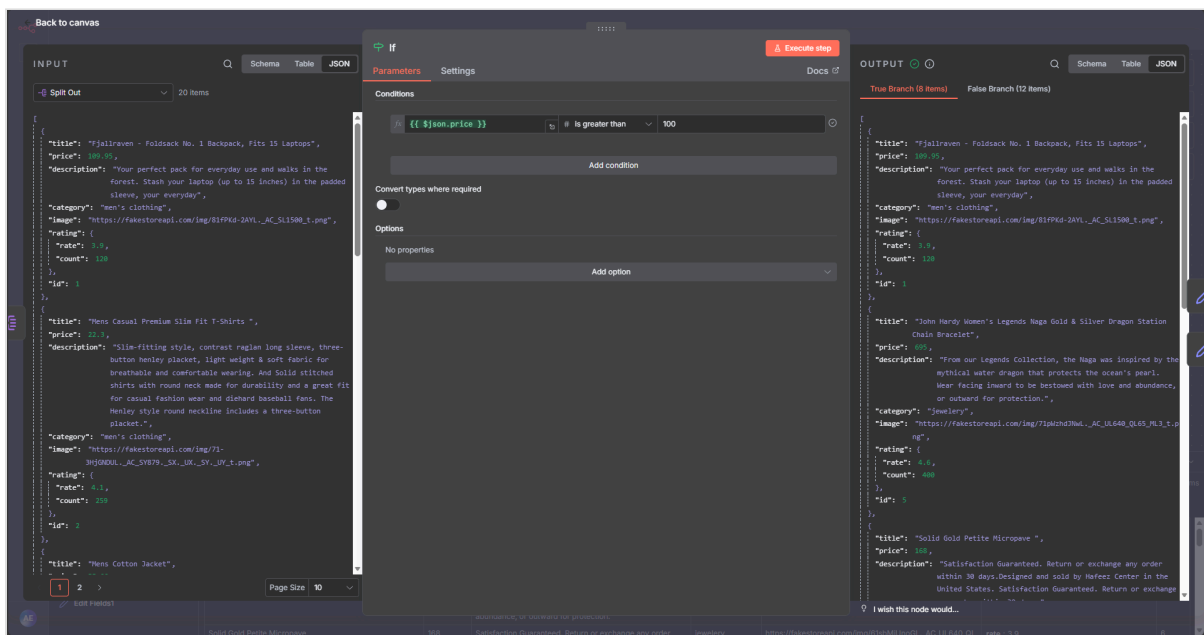
Ejercicio 2: Análisis de Productos (Dificultad: Media)

Este ejercicio consistió en crear un flujo desde cero para analizar productos de la API "Fake Store API". El flujo se inicia con un `Schedule Trigger` (configurado para ejecutarse diariamente) y un `HTTP Request` para obtener la lista completa de productos.

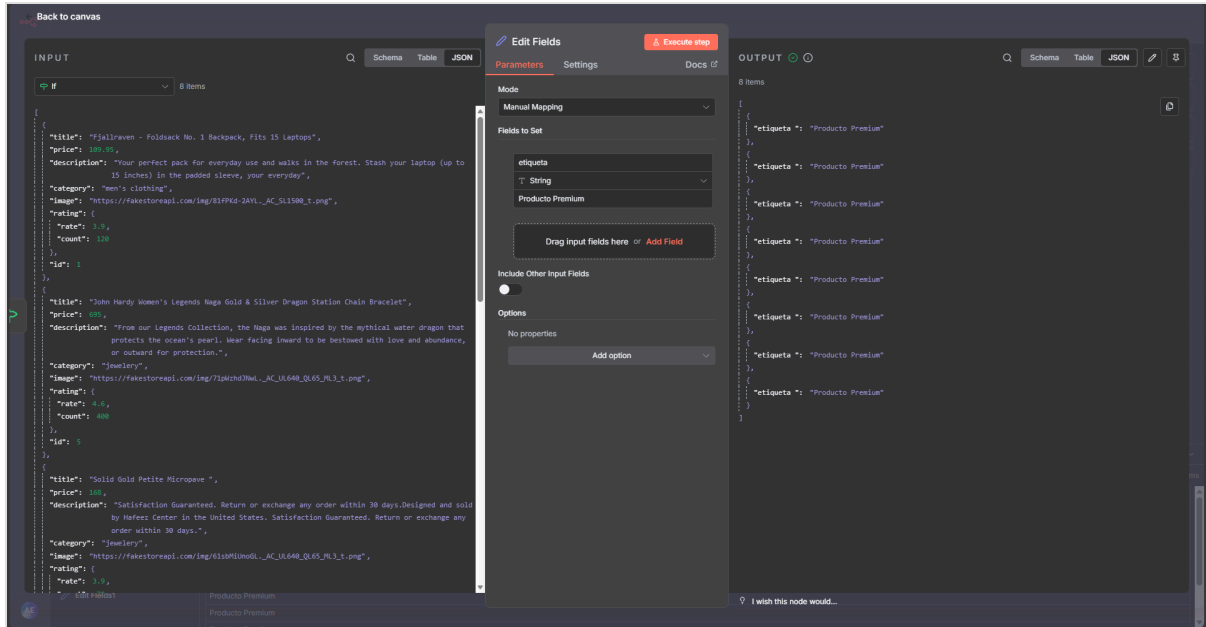
A continuación, se utiliza un nodo `split out` para procesar cada producto de la colección individualmente, tal como se requería.



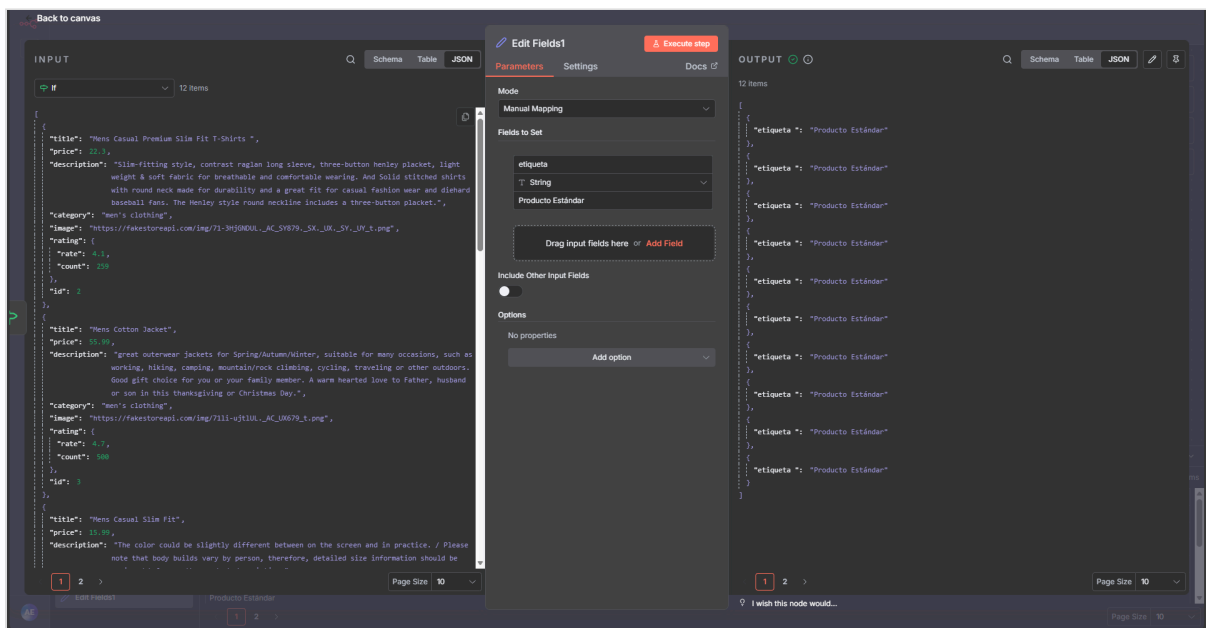
Posteriormente, se implementa la lógica condicional con un nodo IF. Este nodo se configura para comprobar si el campo price de cada producto es 'greater than' (superior a) 100.



Para la rama true (productos con precio superior a 100), se añade un nodo Edit Fields (Set) que crea un nuevo campo llamado etiqueta con el valor "Producto Premium".



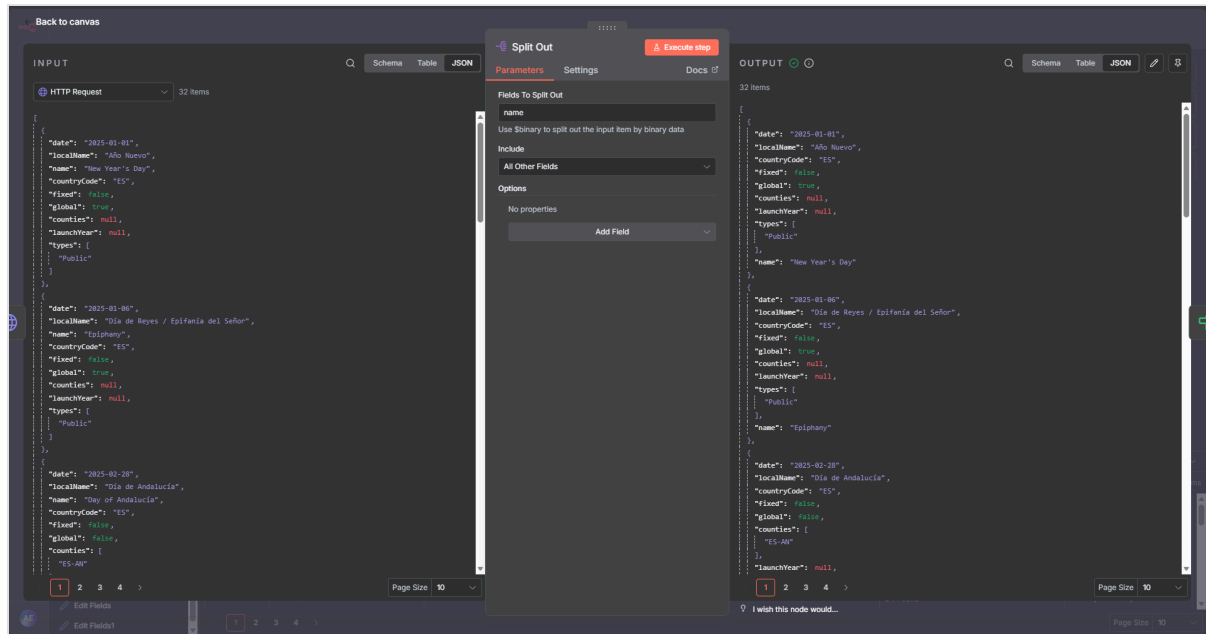
Para la rama false (productos con precio de 100 o inferior), se añade un nodo Edit Fields (Set) similar, pero esta vez asignando el valor "Producto Estándar" al campo etiqueta.



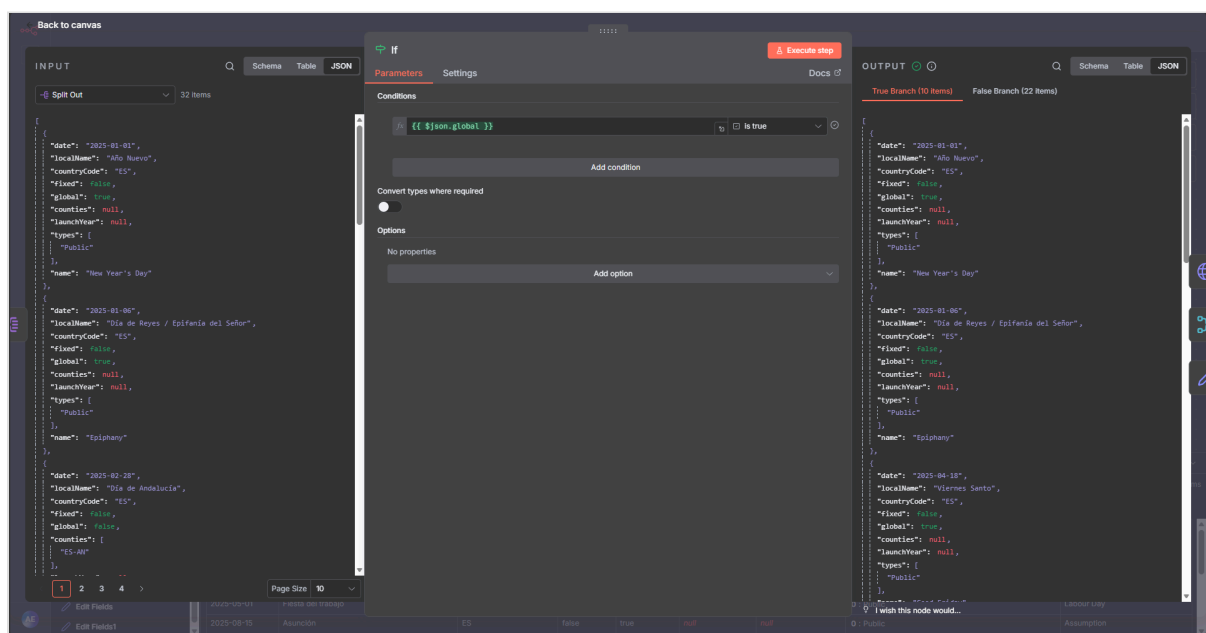
Ejercicio 3: Planificador de Festivos (Dificultad: Alta)

El último ejercicio fue el más avanzado, requiriendo la orquestación de dos APIs. El flujo se inicia con un Manual Trigger y un HTTP Request que obtiene la lista de festivos de España para 2025 desde la API "Nager.Date".

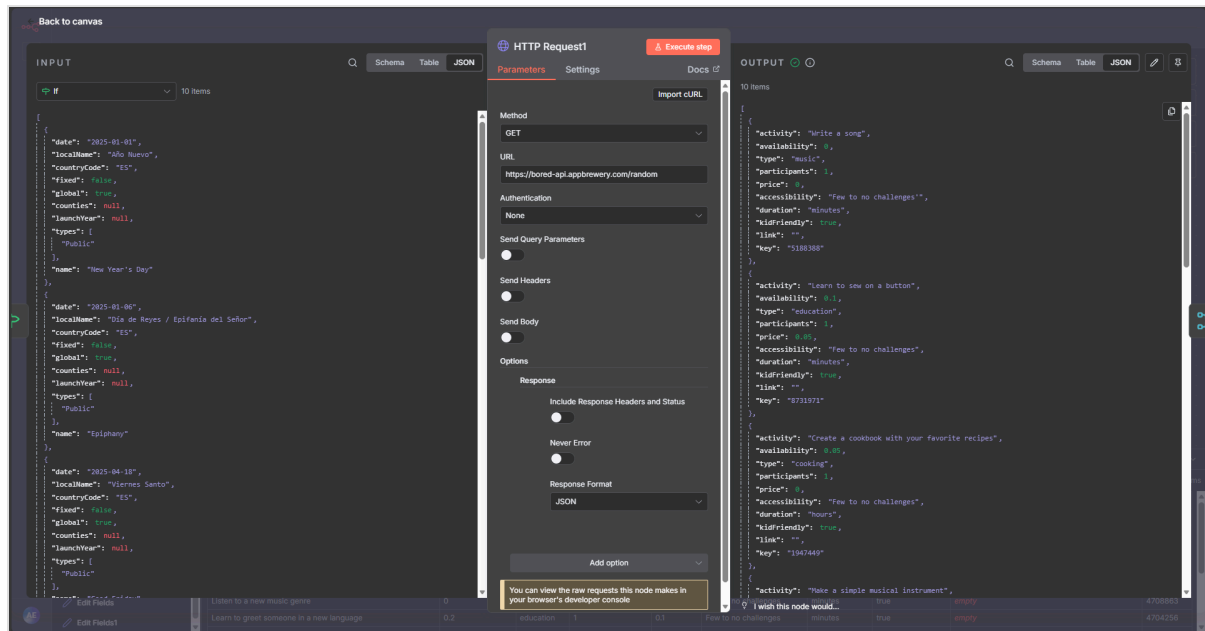
Se utiliza un nodo `split out` para procesar cada festivo de la colección como un ítem individual.



A continuación, un nodo `if` filtra los festivos, comprobando si el campo booleano `global` (que indica si es un festivo nacional) 'is true'.



Para la rama `true` (festivos nacionales), se añade un segundo nodo `HTTP Request`. Este nodo llama a la "Bored API" para obtener una actividad aleatoria.



Finalmente, en la misma rama `true`, se utiliza un nodo `Merge` para combinar los datos del festivo (que vienen de la rama `true` del `IF`) con los datos de la actividad aleatoria (que vienen del segundo `HTTP Request`). Este nodo `Merge` pasa los datos combinados a un nodo `Edit Fields (Set)`, que configura los dos campos finales: `festivo_nombre` (mapeado desde el `name` del festivo) y `actividad_sugerida` (mapeado desde la `activity` de la Bored API), produciendo la salida deseada. La rama `false` (festivos no nacionales) no realiza ninguna acción.

Back to canvas

INPUT

HTTP Request! 10 items

```
{
  "activity": "Write a song",
  "availability": 0,
  "type": "music",
  "participants": 1,
  "price": 0,
  "accessibility": "few to no challenges",
  "duration": "minutes",
  "isfriendly": true,
  "link": "",
  "key": "5188388"
}

{
  "activity": "Learn to sew on a button",
  "availability": 0.1,
  "type": "education",
  "participants": 1,
  "price": 0.85,
  "accessibility": "few to no challenges",
  "duration": "minutes",
  "isfriendly": true,
  "link": "",
  "key": "8731971"
}

{
  "activity": "Create a cookbook with your favorite recipes",
  "availability": 0.85,
  "type": "cooking",
  "participants": 1,
  "price": 0,
  "accessibility": "few to no challenges",
  "duration": "hours",
  "isfriendly": true,
  "link": "",
  "key": "1940449"
}
```

Merge

Parameters Settings Docs

Mode

Combine

Combine By

Position

Number of inputs

2

Options

No properties

Add option

OUTPUT 10 items

```
{
  "activity": "Write a song",
  "availability": 0,
  "type": "music",
  "participants": 1,
  "price": 0,
  "accessibility": "few to no challenges",
  "duration": "minutes",
  "isfriendly": true,
  "link": "",
  "key": "5188388",
  "date": "2020-01-01",
  "localName": "Año Nuevo",
  "countryCode": "ES",
  "fixed": false,
  "global": true,
  "launchYear": null,
  "types": [
    "Public"
  ],
  "name": "New Year's Day"
}

{
  "activity": "Learn to sew on a button",
  "availability": 0.1,
  "type": "education",
  "participants": 1,
  "price": 0.85,
  "accessibility": "few to no challenges",
  "duration": "minutes",
  "isfriendly": true,
  "link": "",
  "key": "8731971",
  "date": "2025-01-06",
  "localName": "Día de Reyes / Epifanía del Señor",
  "countryCode": "ES",
  "fixed": false
}
```