Práctica 8:

DevOps y GitOps - Despliegue y Versionado de Flujos de Trabajo

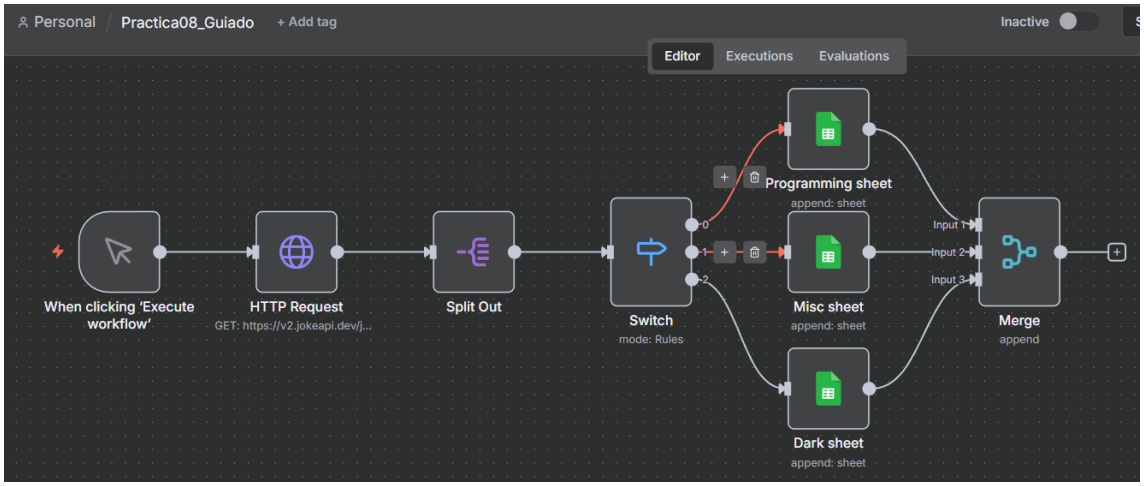
INTEGRACIÓN DE TECNOLOGÍAS Y SERVICIOS INFORMÁTICOS

Daniel Salas Alonso

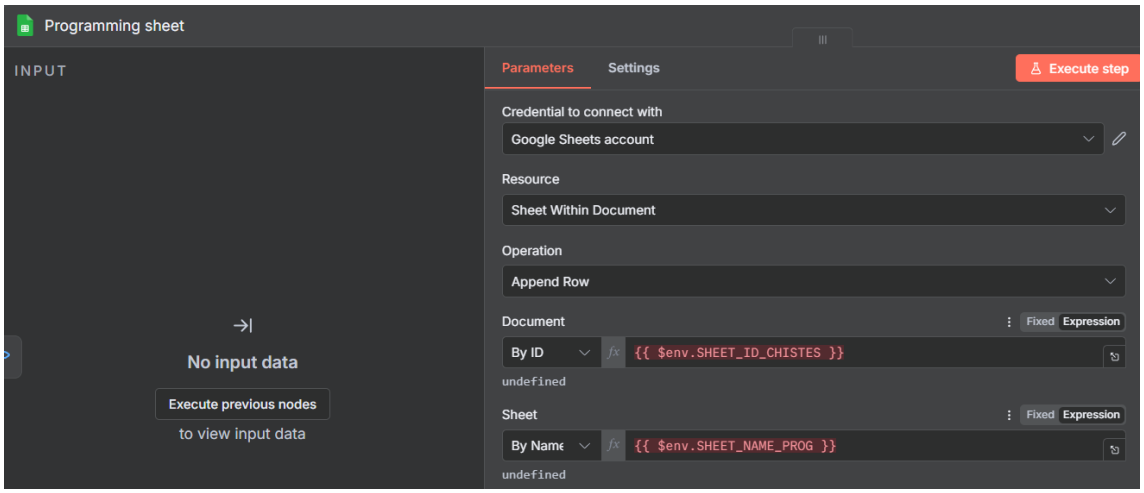
3. Desarrollo del Flujo de Trabajo Guiado

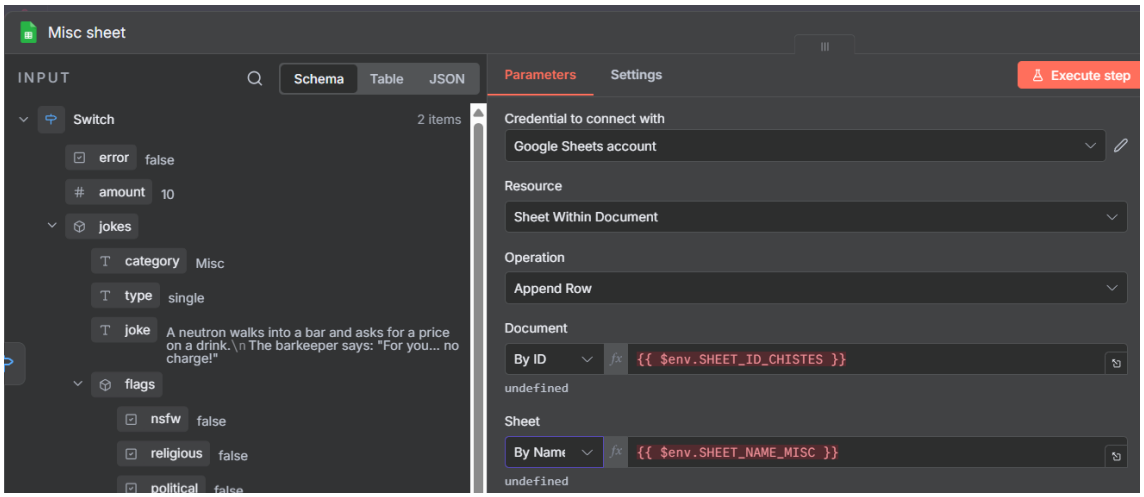
3.1. Paso 1: Refactorizar el Flujo con Variables de Entorno

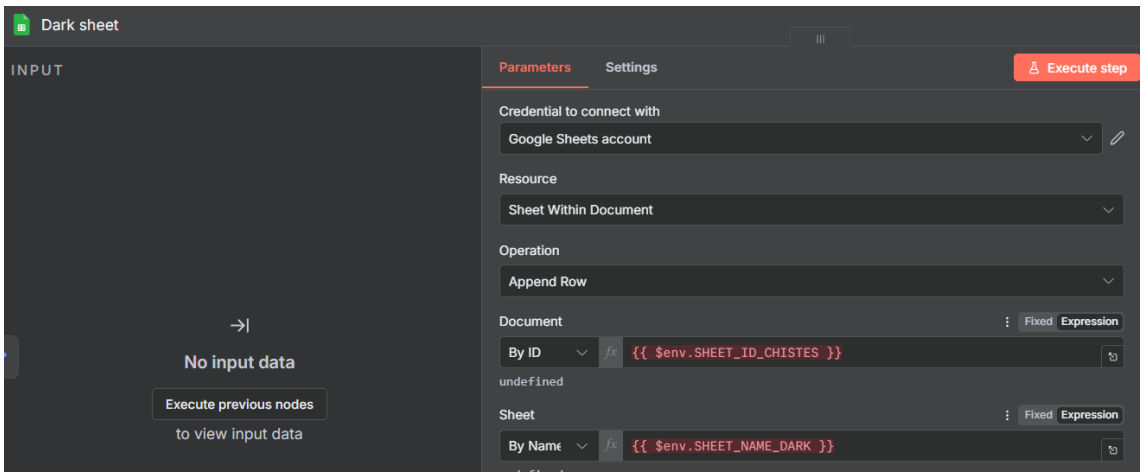
Observamos el flujo de trabajo inicial que vamos a refactorizar. Se trata de un clasificador de chistes (basado en la Práctica 3) que obtiene datos de una API mediante un nodo HTTP Request, los procesa con un Split Out y un Switch, y finalmente los envía a diferentes nodos Google Sheets (Programming, Misc, Dark) según su categoría. El objetivo es desacoplar los IDs de hojas de la lógica del flujo.



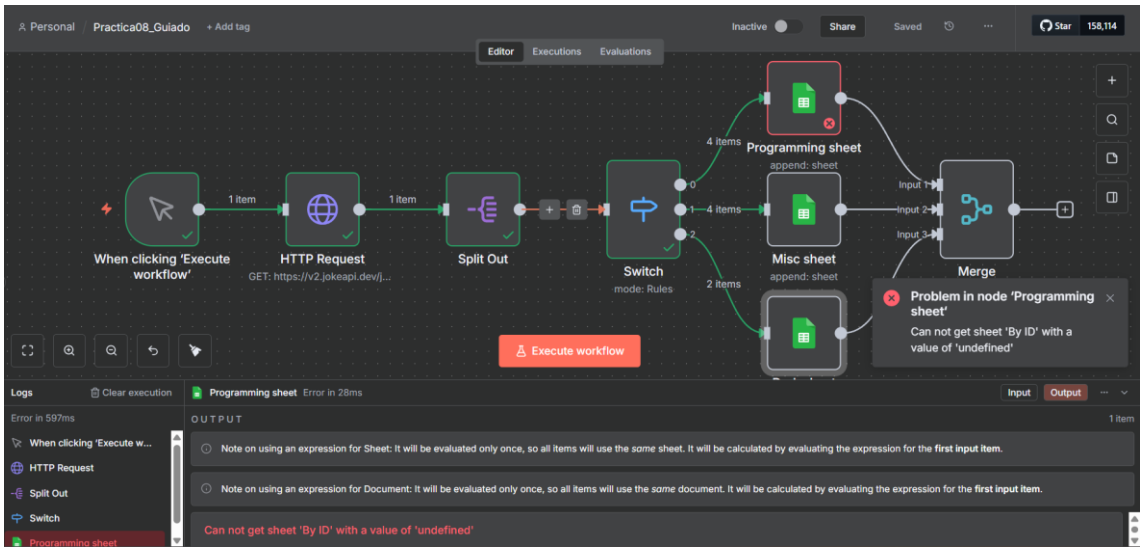
En la configuración del nodo Google Sheets (rama Programming), sustituimos los valores estáticos por expresiones dinámicas. En el campo Document, utilizamos la expresión “$env.SHEET_ID_CHISTES” y en Sheet utilizamos “$env.SHEET_NAME_PROG“. Esto permite que n8n lea estos valores directamente de las variables de entorno del sistema, mejorando la seguridad y portabilidad del flujo. Repetimos el procedimiento para el nodo de la hoja "Misc" y “Dark” cambiando únicamente el nombre de la variable de entorno del nombre de la hoja.







Al intentar ejecutar el flujo sin haber inyectado aún las variables en el contenedor, obtenemos un error en el nodo Google Sheets.



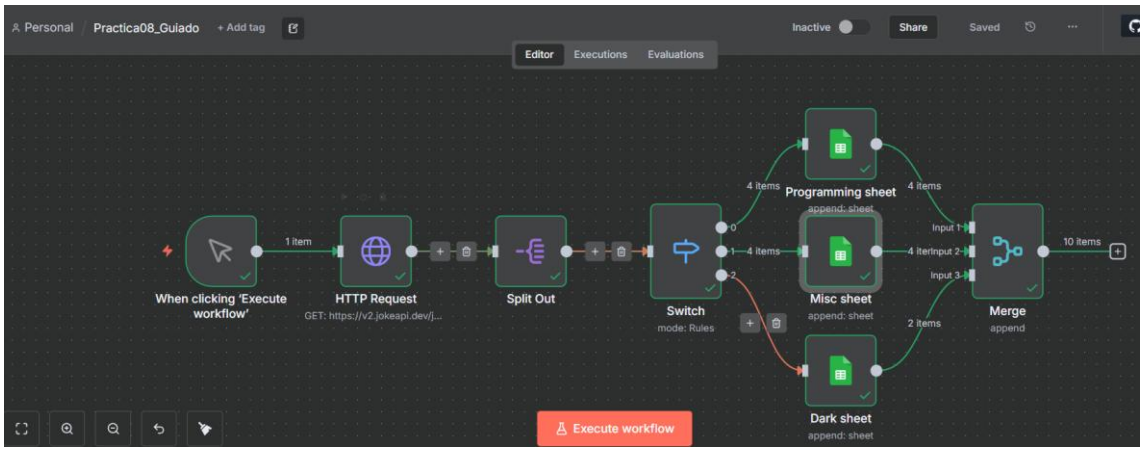
3.2. Paso 2: Inyectar las Variables de Entorno en n8n

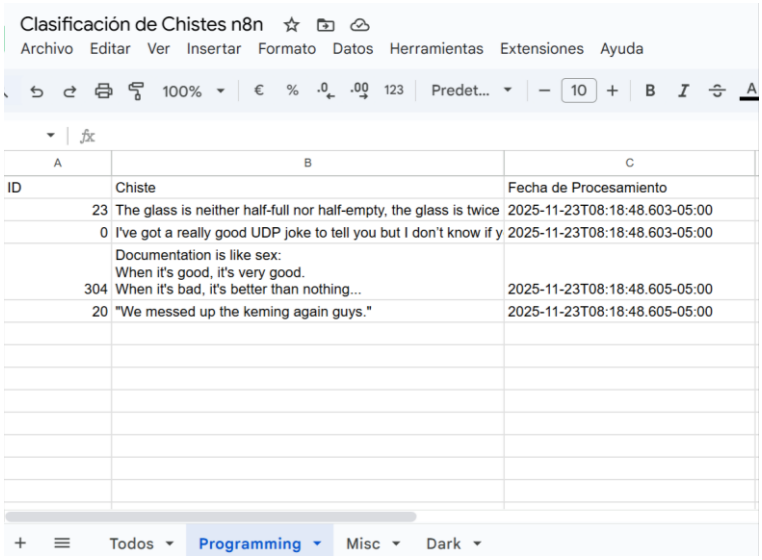
Detenemos la instancia actual y ejecutamos un nuevo contenedor mediante el comando docker run. Utilizamos la bandera -e para inyectar las variables de entorno necesarias: SHEET_ID_CHISTES, SHEET_NAME_PROG, SHEET_NAME_MISC y SHEET_NAME_DARK, asignándoles sus valores reales (IDs y nombres de las hojas de cálculo).

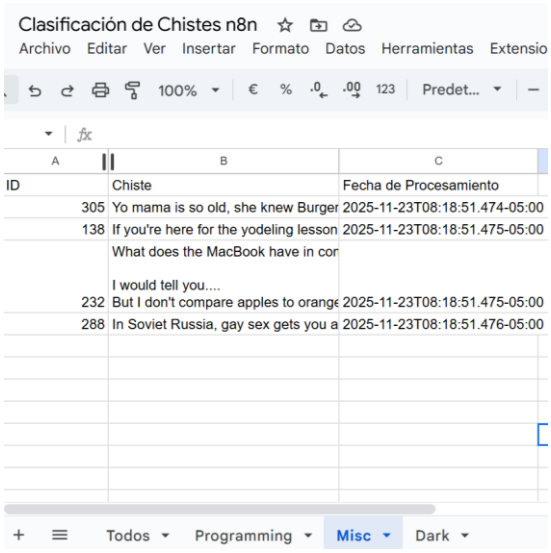
```
dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P01atch
$ docker run -it --rm --name n8n \
    -p 5678:5678 \
    -v \.n8n:/home/node/.n8n \
    -e SHEET_ID_CHISTES="1Cx4gN8PQWN8ReEqymwoL-3EKCxfNNG-RhmhhjJX0YBE" \
    -e SHEET_NAME_PROG="Programming" \
    -e SHEET_NAME_MISC="Misc" \
    -e SHEET_NAME_DARK="Dark" \
    n8nio/n8n
```

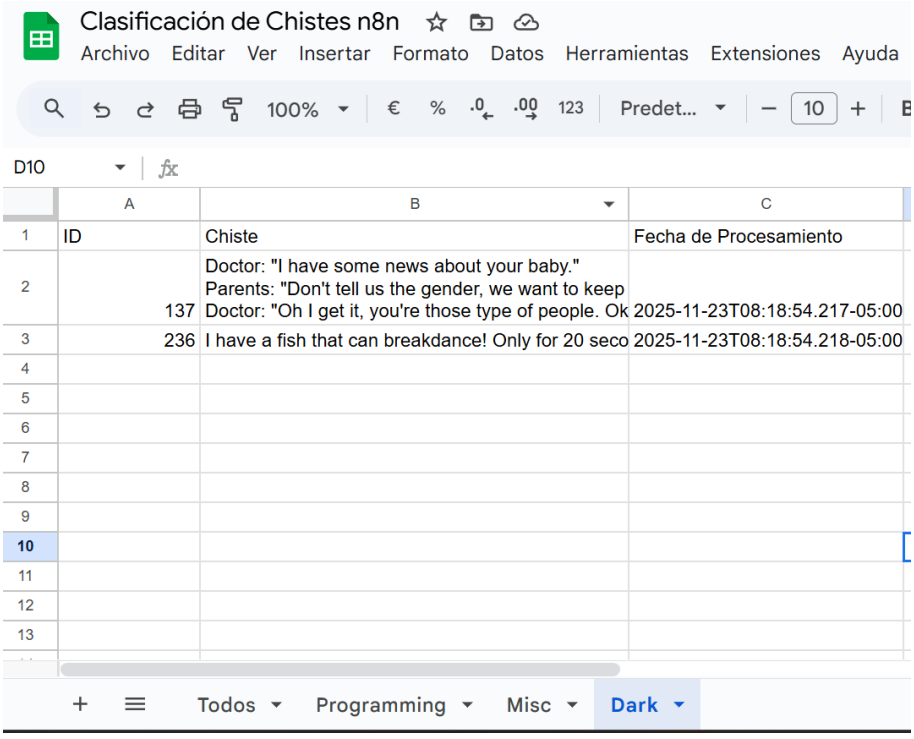
```
[license SDK] Skipping renewal on init: license cert is not due for renewal
Version: 1.120.4

Editor is now accessible via:
http://localhost:5678

Press "o" to open in Browser.
```

Tras reiniciar el contenedor con las variables configuradas, ejecutamos el workflow nuevamente. Observamos que todos los nodos, incluidos los de Google Sheets que antes fallaban, ahora se muestran en verde, indicando que han podido resolver correctamente las expresiones de las variables de entorno. Verificamos el resultado en el documento de Google Sheets.



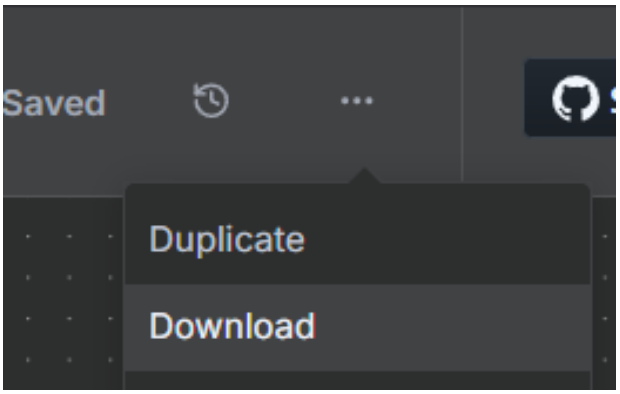

Clasificación de Chistes n8n
Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda
100% € % .0 .00 123 Predet... 10 B I A
fx
A B C
ID Chiste Fecha de Procesamiento
23 The glass is neither half-full nor half-empty, the glass is twice 2025-11-23T08:18:48.603-05:00
0 I've got a really good UDP joke to tell you but I don't know if y 2025-11-23T08:18:48.603-05:00
304 Documentation is like sex:
When it's good, it's very good.
When it's bad, it's better than nothing... 2025-11-23T08:18:48.605-05:00
20 "We messed up the keming again guys." 2025-11-23T08:18:48.605-05:00
Todos Programming Misc Dark


Clasificación de Chistes n8n
Archivo Editar Ver Insertar Formato Datos Herramientas Extensio
100% € % .0 .00 123 Predet...
fx
A B C
ID Chiste Fecha de Procesamiento
305 Yo mama is so old, she knew Burger 2025-11-23T08:18:51.474-05:00
138 If you're here for the yodeling lesson 2025-11-23T08:18:51.475-05:00
What does the MacBook have in cor
I would tell you....
232 But I don't compare apples to orange 2025-11-23T08:18:51.475-05:00
288 In Soviet Russia, gay sex gets you a 2025-11-23T08:18:51.476-05:00
Todos Programming Misc Dark


Clasificación de Chistes n8n
Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda
100% € % .0 .00 123 Predet... 10
D10
fx
A B C
1 ID Chiste Fecha de Procesamiento
2 137 Doctor: "I have some news about your baby."
Parents: "Don't tell us the gender, we want to keep
Doctor: "Oh I get it, you're those type of people. Ok 2025-11-23T08:18:54.217-05:00
3 236 I have a fish that can breakdance! Only for 20 seco 2025-11-23T08:18:54.218-05:00
4
5
6
7
8
9
10
11
12
13
Todos Programming Misc Dark

3.3. Paso 3: Versionar el Flujo de Trabajo en Git

Exportamos el flujo actual. En los tres puntos del workflow, seleccionamos la opción "Download" para guardar el archivo JSON de nuestro flujo de trabajo.

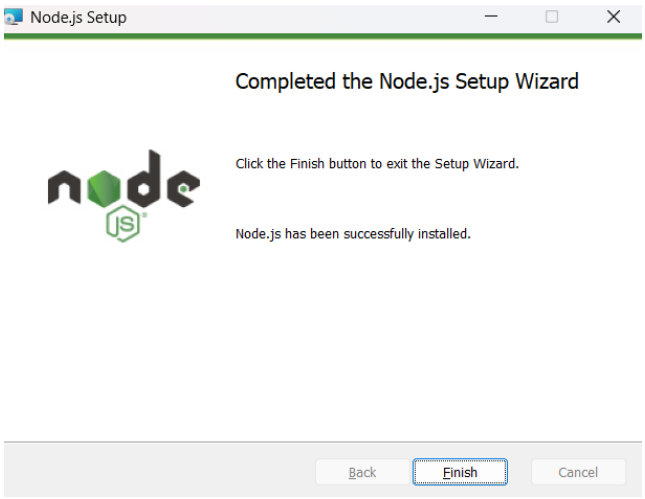


Inicializamos un repositorio local con git init. A continuación, añadimos el archivo exportado (Practica08_Guiado.json) al área de preparación con git add y realizamos el primer git commit con un mensaje descriptivo, estableciendo así el control de versiones del proyecto.

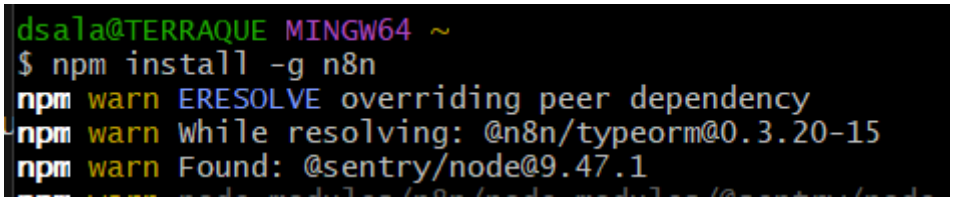
```
dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P08-Guiado
$ git init
Initialized empty Git repository in C:/Users/dsala/repositorios-master/ITSI/P08-
Guiado/.git/

dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P08-Guiado (master)
$ git add Practica08_Guiado.json
warning: in the working copy of 'Practica08_Guiado.json', LF will be replaced by
 CRLF the next time Git touches it

dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P08-Guiado (master)
$ git commit -m "Refactor: Desacoplada configuración de Google Sheets"
[master (root-commit) d0b8e4a] Refactor: Desacoplada configuración de Google She
ets
 1 file changed, 458 insertions(+)
 create mode 100644 Practica08_Guiado.json

dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P08-Guiado (master)
$
```

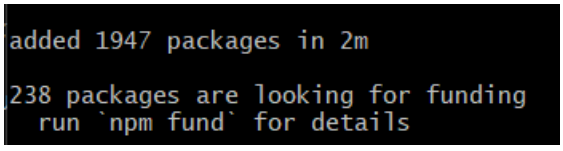
3.4. Paso 4: Simular el Despliegue con n8n-cli

Para simular un entorno de despliegue o ejecutar comandos de administración, procedemos a instalar Node.js.

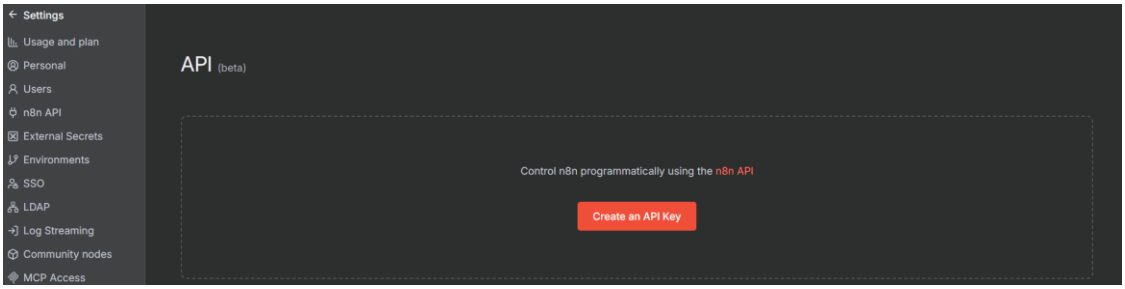


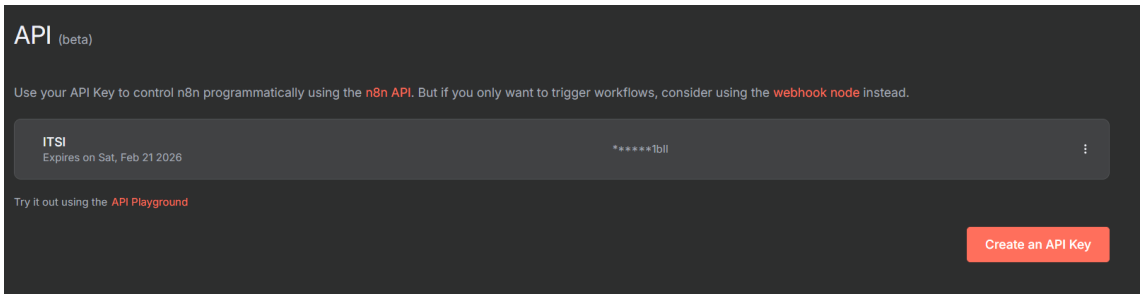
Una vez instalado Node.js, utilizamos el gestor de paquetes npm para instalar n8n de forma global (-g). Esto nos permite utilizar a las funciones CLI (importar/exportar flujos) en el host.



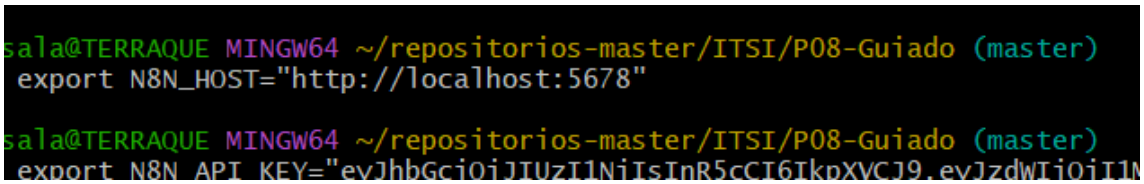


En la interfaz web de n8n, accedemos a "Settings" y "n8n API" para generar una nueva clave de API. Esta API Key servirá como credencial de autenticación para que nuestra herramienta de línea de comandos (CLI) pueda comunicarse de forma segura con la instancia de n8n desplegada.

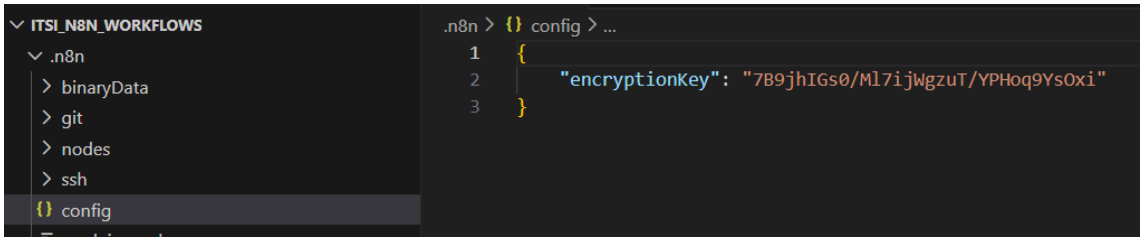


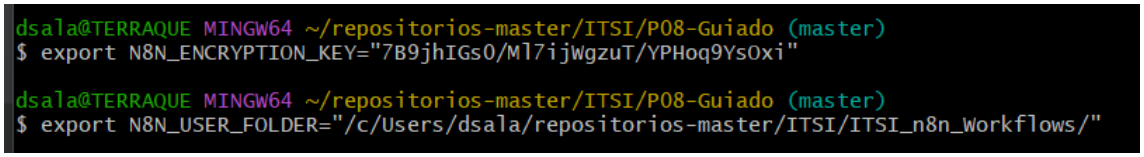


Configuramos las variables de entorno en nuestra terminal local. Exportamos N8N_HOST apuntando a http://localhost:5678 y N8N_API_KEY con el valor obtenido anteriormente.

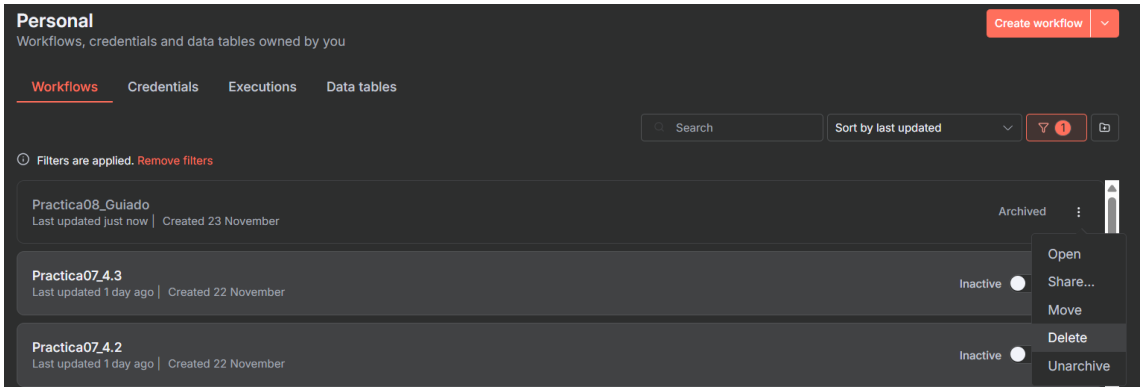


Observamos la configuración de las variables de entorno adicionales necesarias para que la CLI de n8n funcione correctamente con nuestra instancia local y pueda desencriptar las credenciales almacenadas. Exportamos la N8N_ENCRYPTION_KEY con el valor almacenado en el archivo config del volumen y definimos la variable N8N_USER_FOLDER apuntando al directorio local donde residen los datos de configuración y flujos de trabajo (.n8n), asegurando que la herramienta de línea de comandos tenga el contexto necesario para operar.





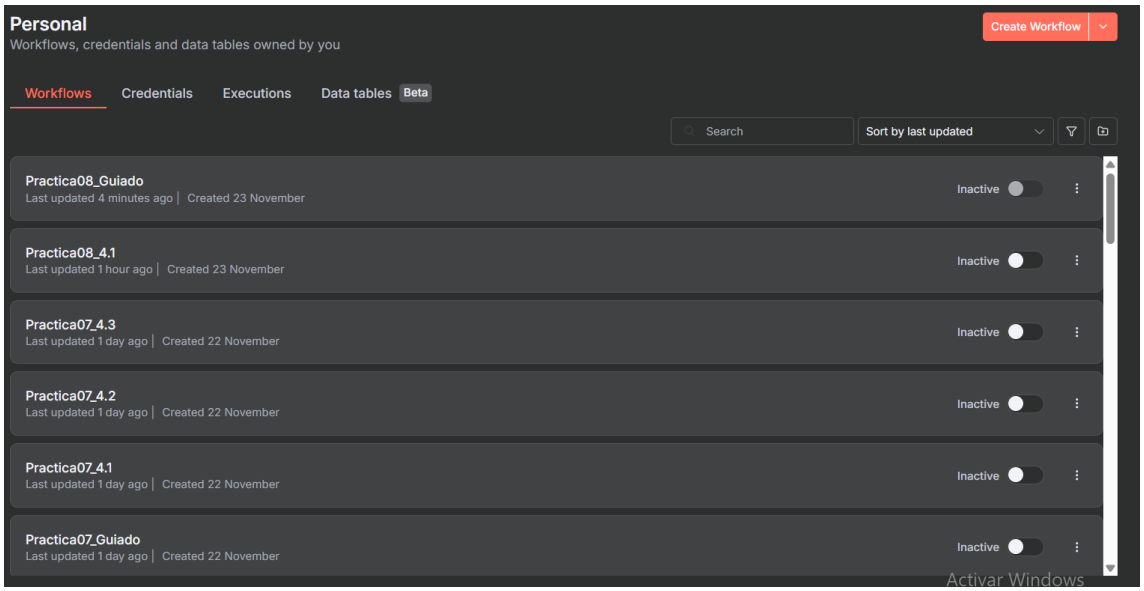
Procedemos a eliminar el workflow "Practica08_Guiado", existente desde la interfaz web de n8n, para importarlo posteriormente.



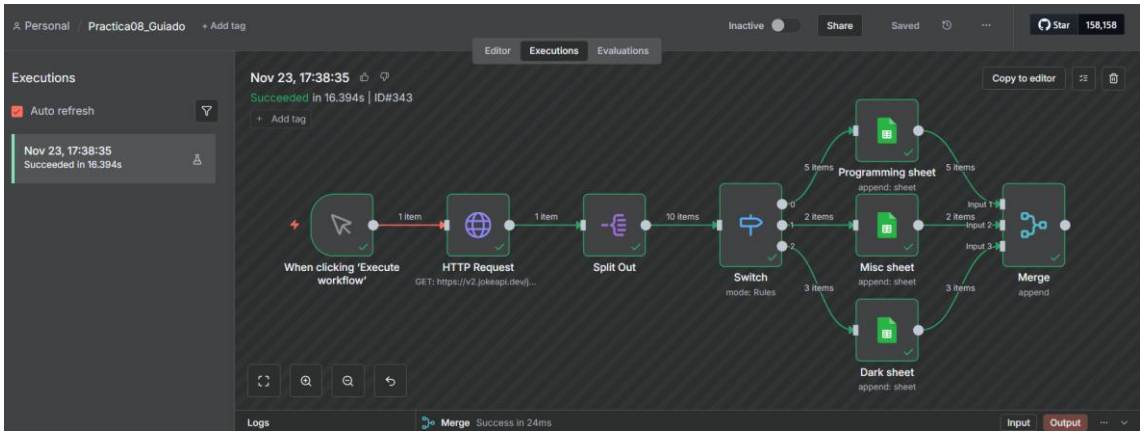
Ejecutamos el comando de importación corrigiendo la sintaxis del flag. Utilizamos n8n import:workflow --input=Practica08_Guiado.json para leer el archivo local (En lugar de --file). De esat manera, conseguimos restaurar el flujo en la instancia utilizando la definición guardada en el archivo JSON.

```
dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P08-Guiado (master)
$ n8n import:workflow --input=Practica08_Guiado.json
Importing 1 workflows...
Could not find workflow
Error: Could not find workflow
    at ActiveWorkflowManager.clearWebhooks (C:\Users\dsala\AppData\Roaming\npm\node_modules\n
8n\src\active-workflow-manager.ts:254:10)
    at ActiveWorkflowManager.remove (C:\Users\dsala\AppData\Roaming\npm\node_modules\n8n\src\
active-workflow-manager.ts:874:4)
    at ImportService.importWorkflows (C:\Users\dsala\AppData\Roaming\npm\node_modules\n8n\src
\services\import.service.ts:82:5)
    at ImportWorkflowsCommand.run (C:\Users\dsala\AppData\Roaming\npm\node_modules\n8n\src\co
mmands\import\workflow.ts:104:3)
    at CommandRegistry.execute (C:\Users\dsala\AppData\Roaming\npm\node_modules\n8n\src\comma
nd-registry.ts:67:4)
    at C:\Users\dsala\AppData\Roaming\npm\node_modules\n8n\bin\n8n:63:2

Could not remove webhooks of workflow "NuU6gODjOrwlnpGL" because of error: "Could not find wo
rkflow"
Successfully imported 1 workflow.
```

Tras refrescar el panel de n8n, confirmamos que el flujo "Practica08_Guiado" aparece en la lista, con la fecha de actualización reciente ("Just now"), validando que el proceso de despliegue mediante CLI ha actualizado la definición del flujo en el servidor.



Ejecutamos manualmente el flujo recién importado para asegurar su integridad operativa. La traza de ejecución muestra todos los nodos en verde, confirmando que la importación mantuvo todas las configuraciones y conexiones correctamente.

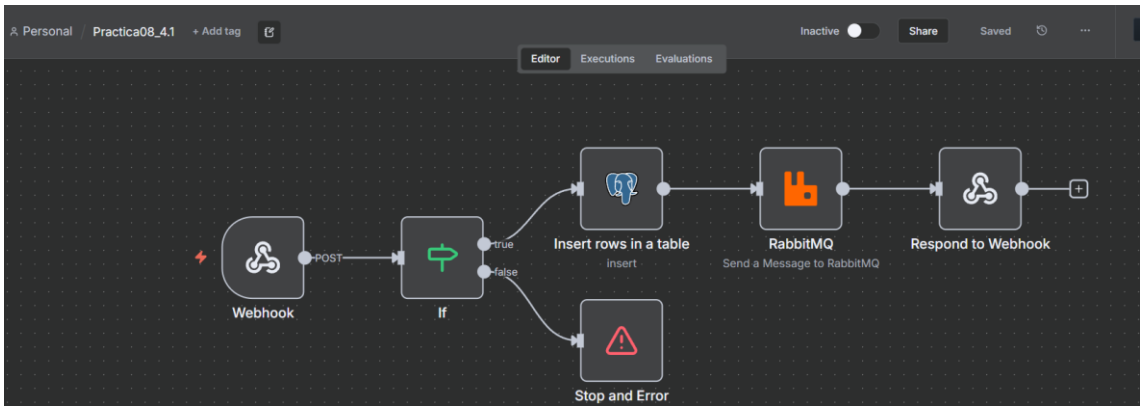


4. Ejercicios Propuestos

4.1. Ejercicio 1: Refactorización Completa (Dificultad: Baja)

1. *Abra el flujo de la Práctica 6 (el que interactuaba con PostgreSQL y RabbitMQ).*

Realizamos una copia del flujo de trabajo de la Práctica 6, que orquesta microservicios insertando datos en PostgreSQL y enviando mensajes a RabbitMQ.

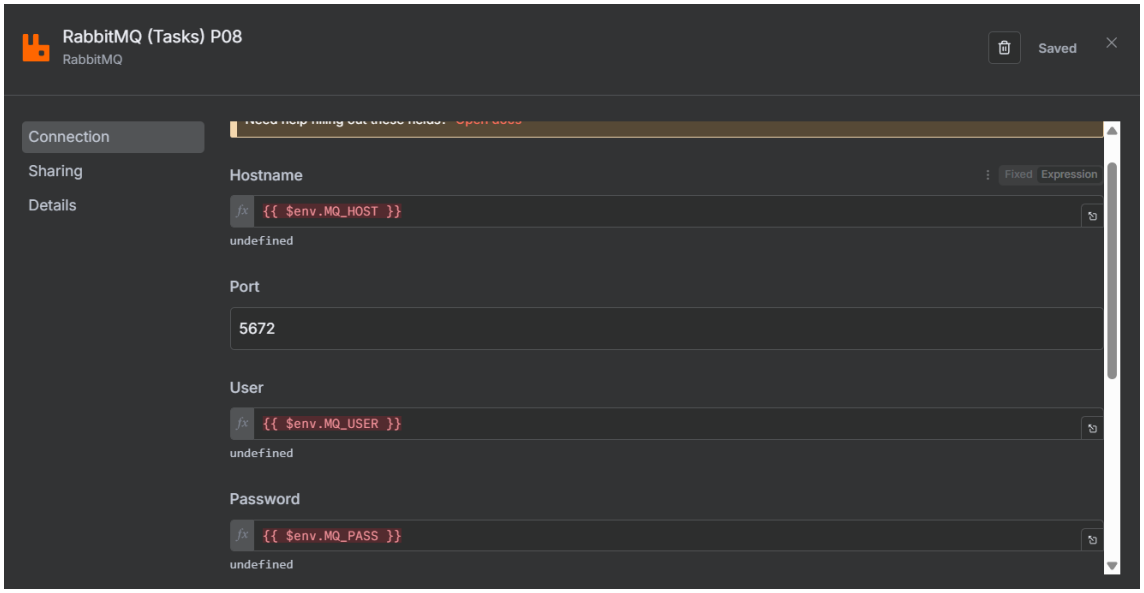


2. *Refactorice todas las credenciales (PostgreSQL y RabbitMQ) para que usen variables de entorno. En lugar de seleccionar la credencial PostgreSQL (Tasks) del desplegable, cree una nueva credencial donde cada campo (Host, Database, User, Pass) sea una expresión.*

Creamos una nueva credencial para PostgreSQL. En lugar de valores estáticos, utilizamos expresiones que apuntan a variables de entorno. Esto hace que la conexión sea dependiente del entorno de despliegue.

3. *Haga lo mismo para la credencial de RabbitMQ.*

De forma análoga, configuramos una nueva credencial para RabbitMQ.



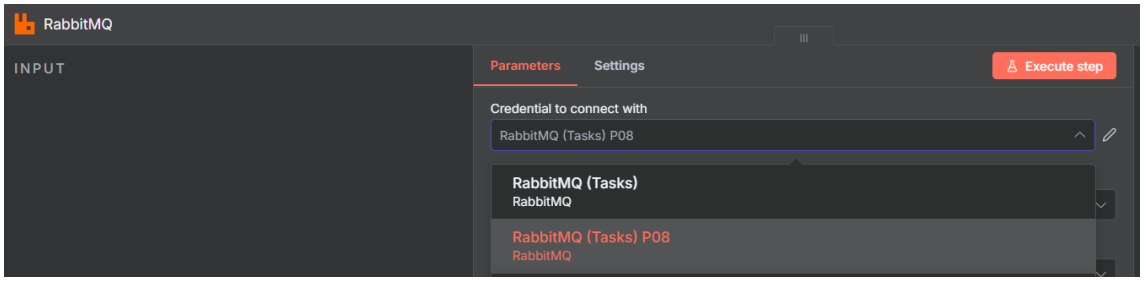
4. *Detenga su contenedor n8n y reinícielo con todos los nuevos flags -e necesarios (-e DB_HOST=db, -e DB_USER=user, etc.).*

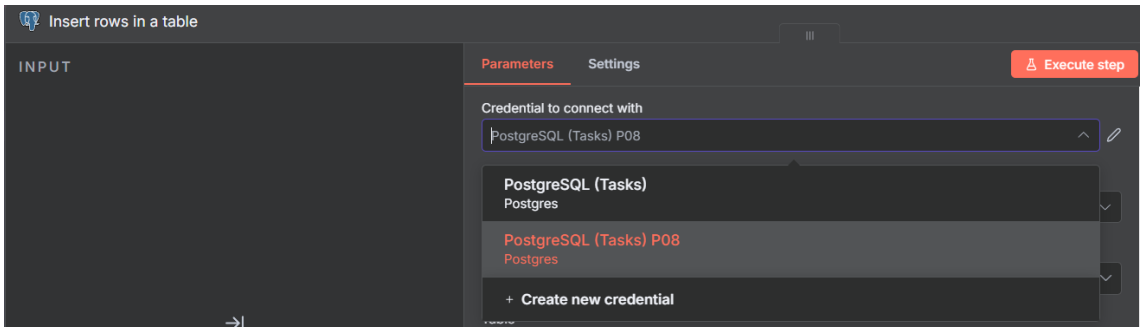
Reiniciamos el contenedor n8n inyectando un conjunto completo de variables de entorno mediante banderas -e. Esto incluye tanto las variables para Google Sheets (SHEET_) como las nuevas variables para la conexión a la base de datos (DB_) y al sistema de colas (MQ_).

```
dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P01
$ docker run -it --rm --name n8n \
    -p 5678:5678 \
    -v \.n8n:/home/node/.n8n \
    -e SHEET_ID_CHISTES="1Cx4gN8PQWN8ReEqymwoL-3EKCxfNNG-RhmhhjJX0YBE" \
    -e SHEET_NAME_PROG="Programming" \
    -e SHEET_NAME_MISC="Misc" \
    -e SHEET_NAME_DARK="Dark" \
    -e DB_HOST="db" \
    -e DB_NAME="taskdb" \
    -e DB_USER="user" \
    -e DB_PASS="password" \
    -e MQ_HOST="mq" \
    -e MQ_USER="guest" \
    -e MQ_PASS="guest" \
    n8nio/n8n
```

```
[license SDK] Skipping renewal on init: license cert is not due for renewal
Version: 1.120.4

Editor is now accessible via:
http://localhost:5678

Press "o" to open in Browser.
```

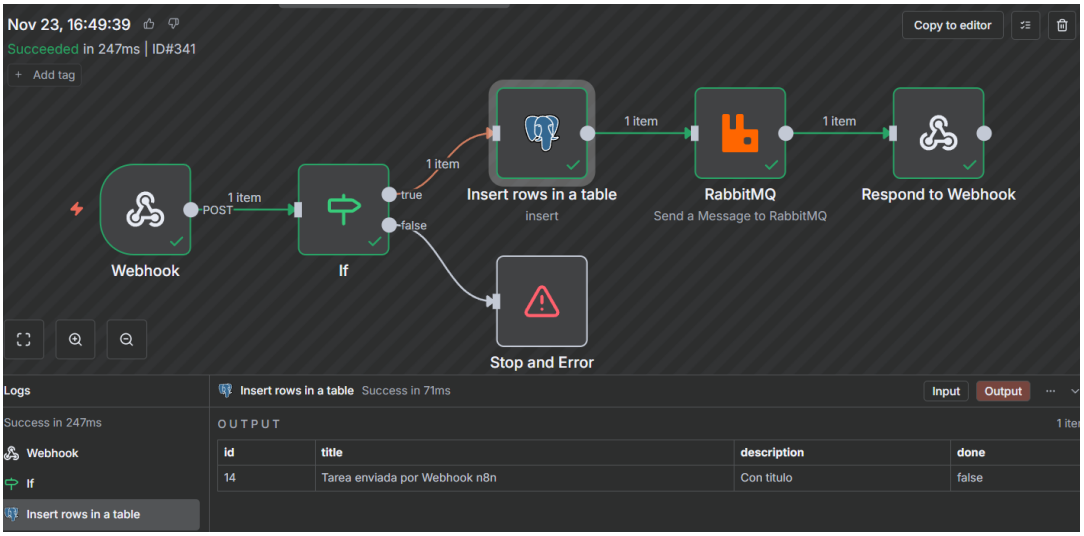
En el nodo RabbitMQ y PostgreSQL del flujo, seleccionamos las nuevas credenciales con las variables de entorno del contenedor que acabamos de crear.





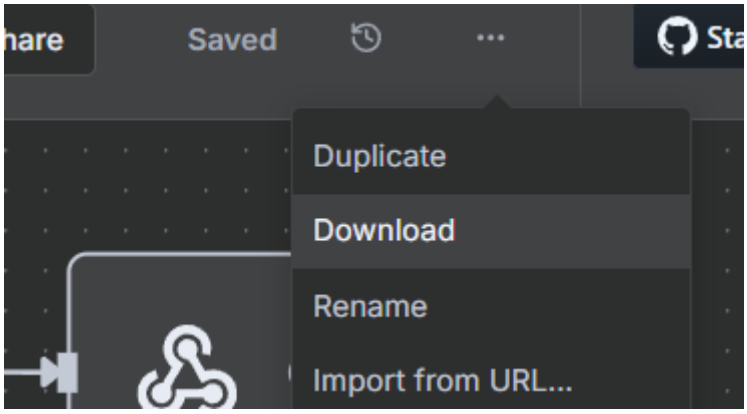
Realizamos una prueba de funcionamiento del flujo enviando una petición curl POST al webhook de entrada y el flujo se ejecuta exitosamente. Esto confirma que n8n ha logrado resolver las variables de entorno, conectarse a la base de datos para insertar el registro y autenticarse en RabbitMQ para encolar el mensaje.

```
dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P05/task-manager-service (main
)
$ curl -X POST -H "Content-Type: application/json" -d "{\"title\": \"Tarea envia
da por Webhook n8n\", \"description\": \"Con titulo\"}" http://localhost:5678/we
bhook-test/f3742d84-3280-4602-a2e6-b7f4e8de3207
{
  "id": 14,
  "title": Tarea enviada por Webhook n8n,
  "description": Con titulo,
  "done": false
}
```



5. *Exporte el flujo refactorizado y añádalo a su repositorio Git.*

Una vez finalizada la refactorización de las credenciales, procedemos a exportar el flujo de trabajo para su versionado.



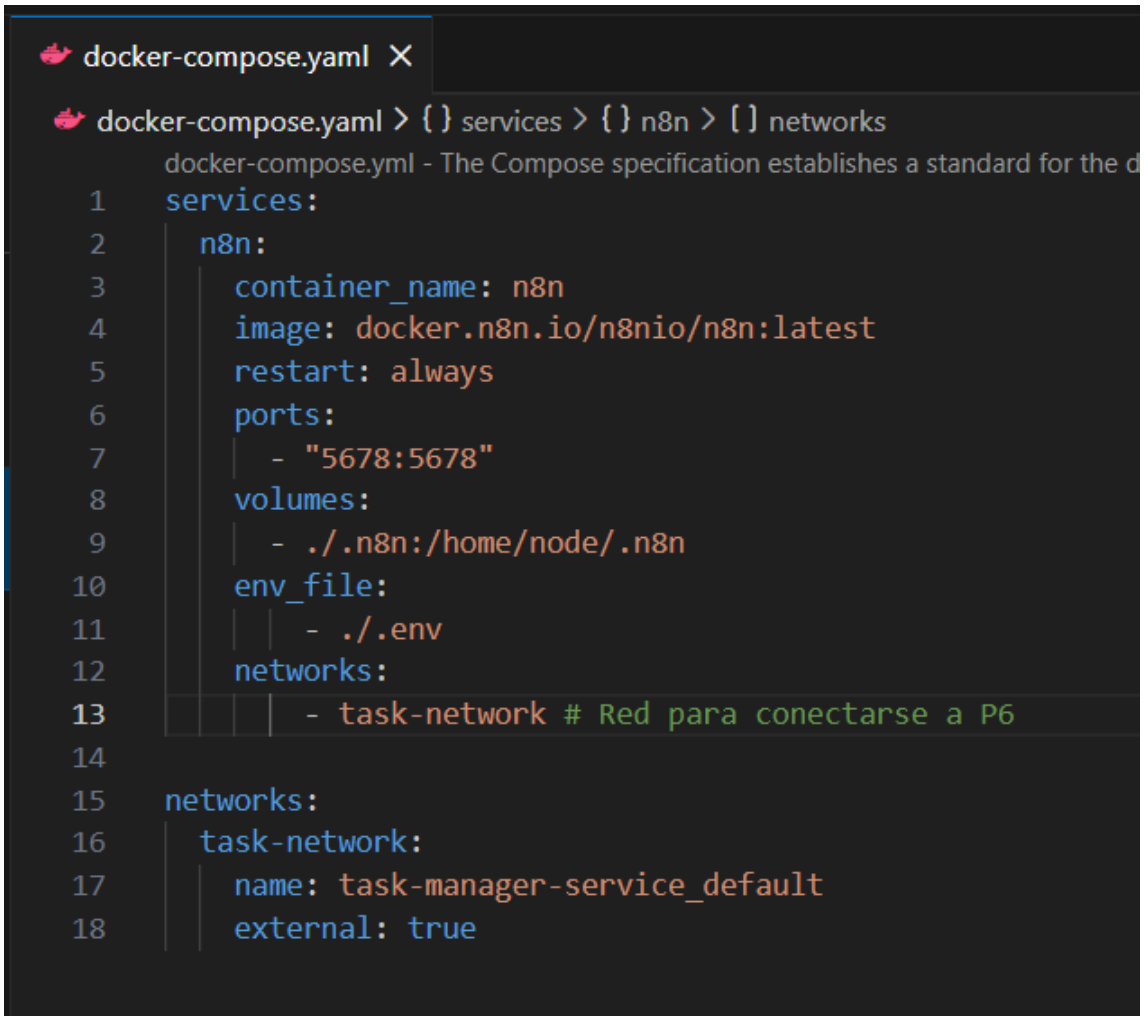
A continuación, integramos el archivo exportado (Practica08_4.1.json) en nuestro sistema de control de versiones local. Ejecutamos el comando git add para añadir el archivo al área de preparación, seguido de git commit.

```
dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P08-Guiado (master)
$ git add Practica08_4.1.json
warning: in the working copy of 'Practica08_4.1.json', LF will be replaced by CR
LF the next time Git touches it

dsala@TERRAQUE MINGW64 ~/repositorios-master/ITSI/P08-Guiado (master)
$ git commit -m "Refactor: Postgress y RabbitMQ"
[master 8c8436d] Refactor: Postgress y RabbitMQ
 1 file changed, 261 insertions(+)
 create mode 100644 Practica08_4.1.json
```

4.2. Ejercicio 2: docker-compose y .env (Dificultad: Media)

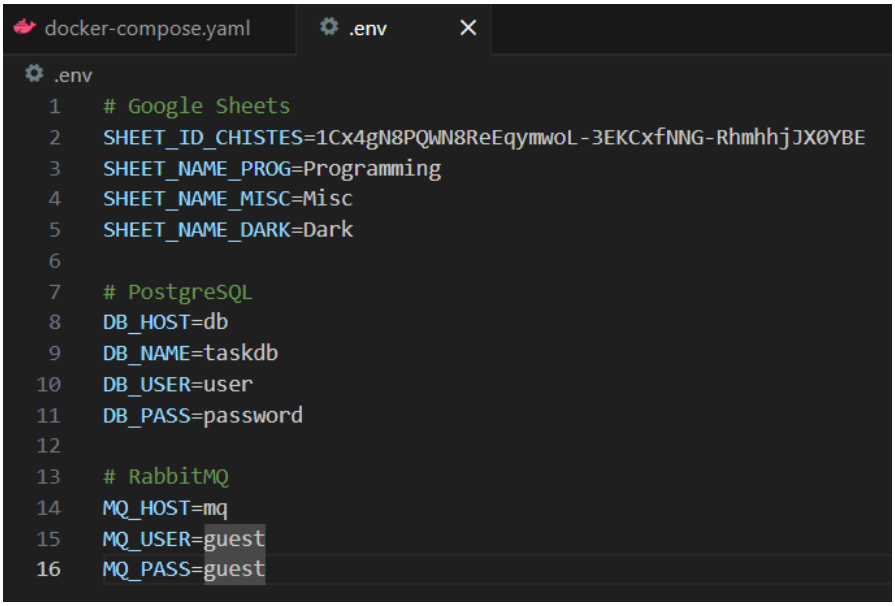
1. *Cree un fichero docker-compose.yml para su instancia de n8n.*

Creamos un archivo Docker compose para definir la infraestructura de n8n de forma declarativa. Definimos el servicio n8n, mapeamos el puerto 5678, configuramos el volumen persistente y utilizamos la directiva env_file: ./.env para cargar las variables. También conectamos el contenedor a la red externa task-manager-service_default para que tenga visibilidad de los microservicios.



2. *Cree un fichero .env (en el mismo directorio) y ponga todas sus variables de entorno allí:*

Creamos el archivo .env donde centralizamos todas las variables de entorno que especificamos en el comando “run” anterior.

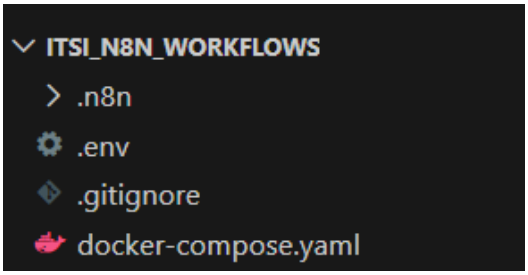


3. *Cree un fichero .gitignore y añada .env a él (¡los secretos NUNCA van a Git!).*

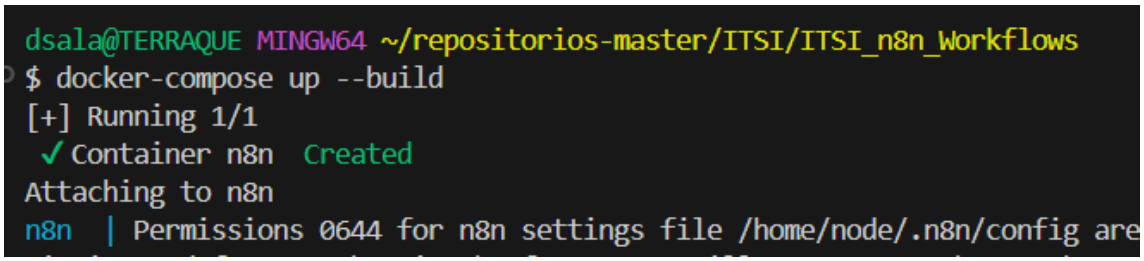
Configuramos el archivo .gitignore incluyendo .env para que las variables de entorno no se suban al repositorio de control de versiones.

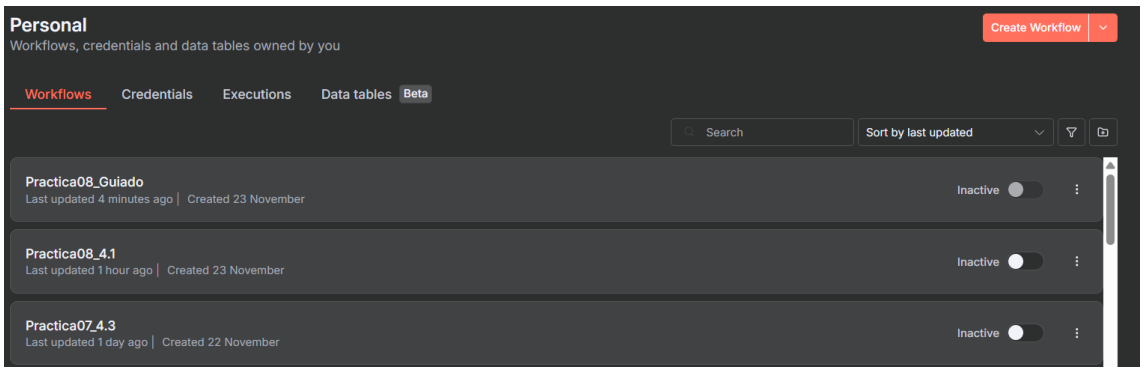
4. *Detenga su contenedor docker run e inicie con docker-compose up -d. Su n8n se iniciará con todas las variables de entorno cargadas desde el fichero .env.*

Esta es la estructura que nos queda en nuestro proyecto n8n al haber creado los archivos anteriores.

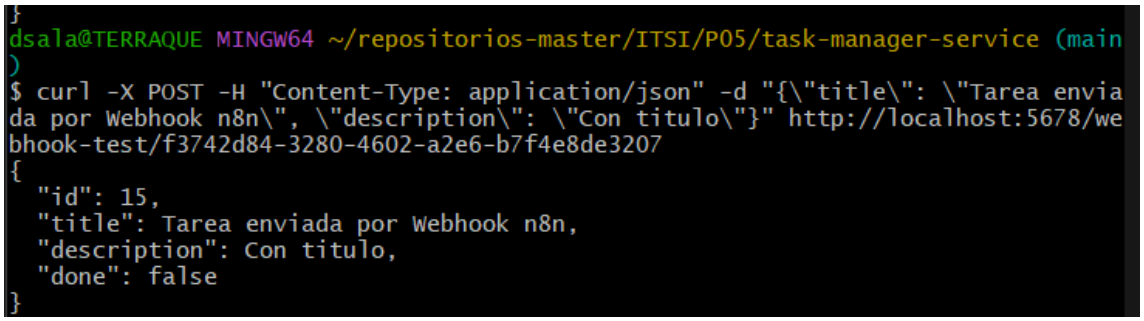


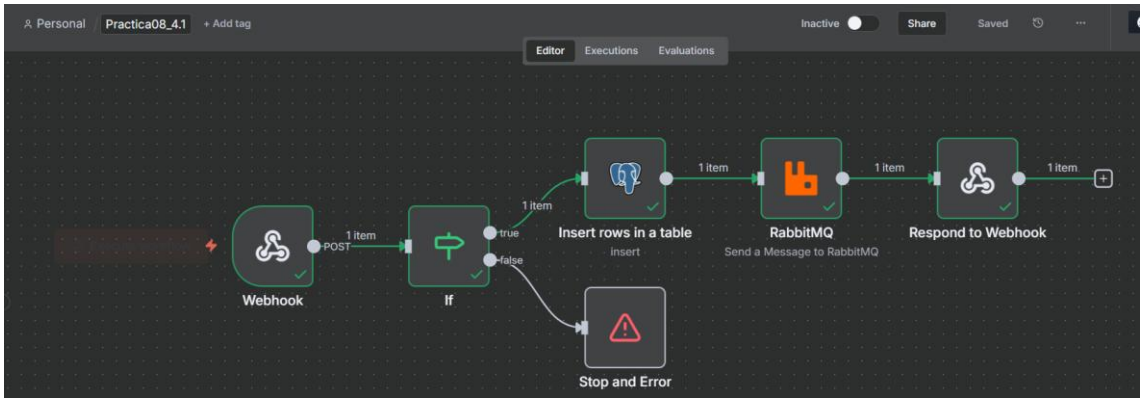
Ejecutamos el Docker compose y accedemos a la interfaz de n8n. Verificamos que se detecta el volumen correctamente, ya que nuestros flujos de trabajo están disponibles.





Realizamos una nueva prueba funcional mediante curl en el workflow anterior y Confirmamos la ejecución correcta del flujo en el entorno orquestado por Docker compose, donde se añade n8n a la network de task_manager_services y se detectan todas las variables de entorno.

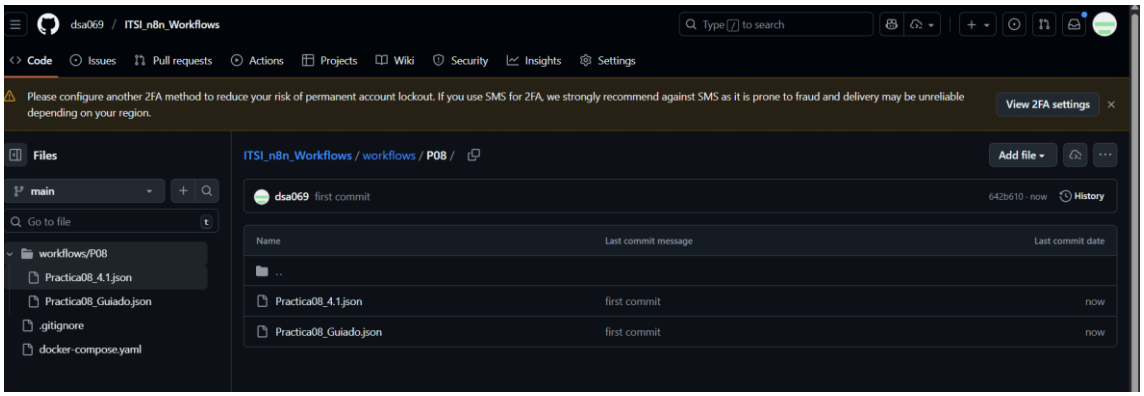




4.3. Ejercicio 3: Escribir un Pipeline de CI/CD (Dificultad: Alta - Conceptual)

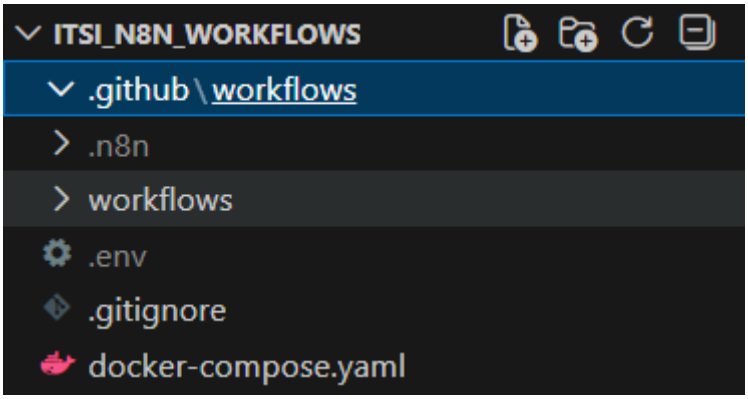
1. *Asuma que ha subido su repositorio (con workflow.json y el docker-compose.yml del E2) a GitHub.*

Subimos nuestro proyecto n8n a un repositorio de GitHub. Contiene la carpeta workflows con los archivos JSON y los archivos de configuración de Docker.



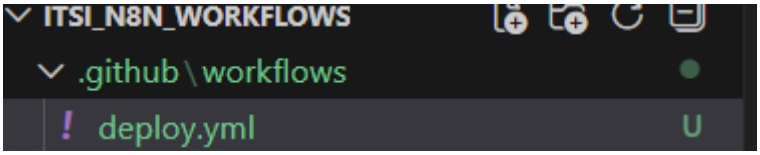
2. *Cree un directorio .github/workflows/ en su proyecto.*

Creamos la estructura de directorios .github/workflows/ en el proyecto.



3. *Dentro, cree un fichero deploy.yml.*

Creamos el archivo deploy.yml en el sistema de carpetas añadido anteriormente.



4. *Escriba el YAML para un GitHub Action.*

Definimos el archivo deploy.yml. Especificamos el archivo JSON de nuestro workflow, donde se configura para dispararse ante un push en la rama main. Sus pasos incluyen: hacer checkout del código, instalar Node.js, instalar n8n-cli globalmente y finalmente ejecutar el comando de importación de n8n utilizando los secretos N8N_HOST y N8N_API_KEY almacenados.

```yaml
# deploy.yml M
# .github > workflows > deploy.yml > { } on > { } push > [ ] paths
# GitHub Workflow - YAML GitHub Workflow (github-workflow.json)
name: Deploy n8n Workflow

on:
  push:
    branches:
      - main
    paths:
      - 'workflows/P08/Practica08_Guiado.json' # Solo se ejecuta si el flujo cambia

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v3

      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'

      - name: Install n8n-cli
        run: npm install n8n -g

      - name: Deploy to Production
        env:
          N8N_HOST: ${{ secrets.N8N_PROD_HOST }}
          N8N_API_KEY: ${{ secrets.N8N_PROD_API_KEY }}
          # (Importante: usar el ID para actualizar, no crear uno nuevo)
        run: |
          n8n import:workflow --file=workflows/P08/Practica08_Guiado.json --id=NuU6gODjOrwlnpGL
```

Realizamos el commit y push del archivo de configuración del pipeline. Al subir este archivo a GitHub, la plataforma debería detectar la acción y, si se cumplen las condiciones (push a main), iniciará el proceso de despliegue automático definido.