# **📦 Curso: Despliegue de Aplicaciones con Docker**

🎓 Información del Curso

Curso: Despliegue de Aplicaciones con Docker  
Proyecto: Práctica de Docker Compose  
Profesor: Ing. Edison Naranjo (CEC-EPN)  
Fecha: 21 de Septiembre de 2025

# **🚀 Tarea Final Flowise – Grupo 1 (Municipio de Quito)**

Bienvenido al repositorio del proyecto **Flowise**, desarrollado por el **Grupo 1 del Municipio de Quito**.  
 Este proyecto forma parte del **Trabajo Final: Despliegue de entornos de automatización con Flowise integrados con PostgreSQL en Docker Compose**.

## **👥 Integrantes**

|  |  |
| --- | --- |
| **Nombre completo** | **Repositorio GitHub** |
| Carpio Zaquinaula, Byron Orlando | <https://github.com/bcarpio16/flowise.git> |
| Villarroel Vera, Milton Orlando | <https://github.com/movillarroel/flowise.git> |
| Mena Segura, Edison Fabián | <https://github.com/Bulls1168-C/flowise.git> |
| Benavides Freire, Alex Vicente | <https://github.com/abenavides86/flowise.git> |
| Gallardo Nicolalde, Marcelo Iván | <https://github.com/panivinux/flowise.git> |

## **📄 Descripción General**

El objetivo de esta parte del trabajo es desplegar **Flowise**, integrado con **PostgreSQL**, utilizando **Docker Compose**.  
 Se busca aplicar conceptos de **orquestación de contenedores**, **persistencia de datos**, **separación de servicios** y **buenas prácticas** en la gestión de entornos. Como valor agregado se ha instalado certificados autofirmados SSL y el ingreso al sistema se realizará en [https://iaflujos.quito.gob.ec](https://iaflujos.quito.gob.ec/)

## **🔧 Requisitos Técnicos para Flowise**

1️⃣ **Infraestructura**

* Docker y Docker Compose instalados en el host.
* El contenedor de Flowise debe comunicarse a través de una **red personalizada** en Docker.
* La base de datos PostgreSQL debe tener un **volumen persistente** para guardar los datos.

2️⃣ **Servicios – Flowise + PostgreSQL**

* Flowise disponible en el **puerto 3000**.
* Base de datos PostgreSQL dedicada e independiente de otros servicios (como n8n).
* Uso de variables de entorno definidas en un archivo .env.

3️⃣ **Buenas Prácticas**

* Usar un archivo .env para todas las credenciales y configuraciones sensibles.
* Mantener separación clara de servicios dentro del docker-compose.yml.
* No usar bind mounts, garantizando portabilidad de la aplicación.
* Documentar en README.md instrucciones claras para levantar y detener Flowise.

## **⚡ Instalación y primer arranque**

### **La instalación de Flowise esta configurado con certificados autofirmados SSL y su validación se realizará en esta URL** [**https://iaflujos.quito.gob.ec**](https://iaflujos.quito.gob.ec/)**, para esto seguir los siguientes pasos:**

1️⃣ **Clonar el repositorio:**

git clone <https://github.com/panivinux/flowise.git>cd flowise

📦 Estructura del proyecto

tree -a

flowise/  
├── app/ # Código fuente de la aplicación  
├── certs/ # Certificados SSL  
├── docker-compose.yml # Configuración de Docker Compose  
├── README.md # Documentación del proyecto  
└── .env # Variables de entorno de ejemplo

2️⃣ Detalle de docker-compose.yml  
networks:  
 flowise-net:  
 driver: bridge  
  
volumes:  
 flowise\_postgres\_data:  
  
services:  
 # ─── PostgreSQL para Flowise ───  
 flowise\_postgres:  
 image: postgres:15  
 container\_name: flowise\_postgres  
 environment:  
 POSTGRES\_USER: ${DATABASE\_USER}  
 POSTGRES\_PASSWORD: ${DATABASE\_PASSWORD}  
 POSTGRES\_DB: ${DATABASE\_NAME}  
 networks:  
 - flowise-net  
 ports:  
 - "5432:5432"  
 volumes:  
 - flowise\_postgres\_data:/var/lib/postgresql/data  
 restart: unless-stopped  
  
 # ─── Flowise ───  
 flowise:  
 image: flowiseai/flowise:latest  
 container\_name: flowise  
 env\_file:  
 - .env  
 depends\_on:  
 - flowise\_postgres  
 networks:  
 - flowise-net  
 expose:  
 - "3000"  
 restart: unless-stopped  
  
 # ─── Nginx Reverse Proxy ───  
 nginx:  
 image: nginx:alpine  
 container\_name: flowise\_nginx  
 restart: always  
 ports:  
 - "80:80"  
 - "443:443"  
 volumes:  
 - ./nginx.conf:/etc/nginx/conf.d/default.conf:ro  
 - ./certs:/etc/nginx/certs:ro  
 depends\_on:  
 - flowise  
 networks:  
 - flowise-net

3️⃣ Configurar variables de entorno (opcional):

Si deseas cambiar el usuario y clave, lo puedes realizar en este archivo .env de igual manera los parámetros de la base de datos:

nano .env

# ─── Credenciales iniciales (solo para primer arranque) ───  
FLOWISE\_USERNAME=admin  
FLOWISE\_PASSWORD=Adm1n!2025#  
# ─── Configuración de la base de datos ───  
DATABASE\_TYPE=postgres  
DATABASE\_HOST=flowise\_postgres  
DATABASE\_PORT=5432  
DATABASE\_NAME=flowisedb  
DATABASE\_USER=flowiseusr  
DATABASE\_PASSWORD=DbUsr!2025#XyZ  
# ─── API Keys ───  
HUGGINGFACE\_API\_KEY=hf\_cnBIPkkTjInCVCyUkWxaItYvhlmDQKvkUj  
# ─── Puerto de la aplicación ───  
PORT=3000

4️⃣ Levantar los servicios con Docker Compose:

docker compose up -d

🖥️ Configuración de dominio local iaflujos.quito.gob.ec

1️⃣ Obtener la IP del Gateway del contenedor flowise:

docker inspect -f '{{range .NetworkSettings.Networks}}{{.Gateway}}{{end}}' flowise

2️⃣ Editar /etc/hosts como superusuario:

sudo nano /etc/hosts

Agregar al final: Ejemplo:

172.18.0.1 iaflujos.quito.gob.ec

Tú escribirar en el archivo hosts la IP del Gateway que salio del contendor flowise.

3️⃣ Verificar resolución del dominio:

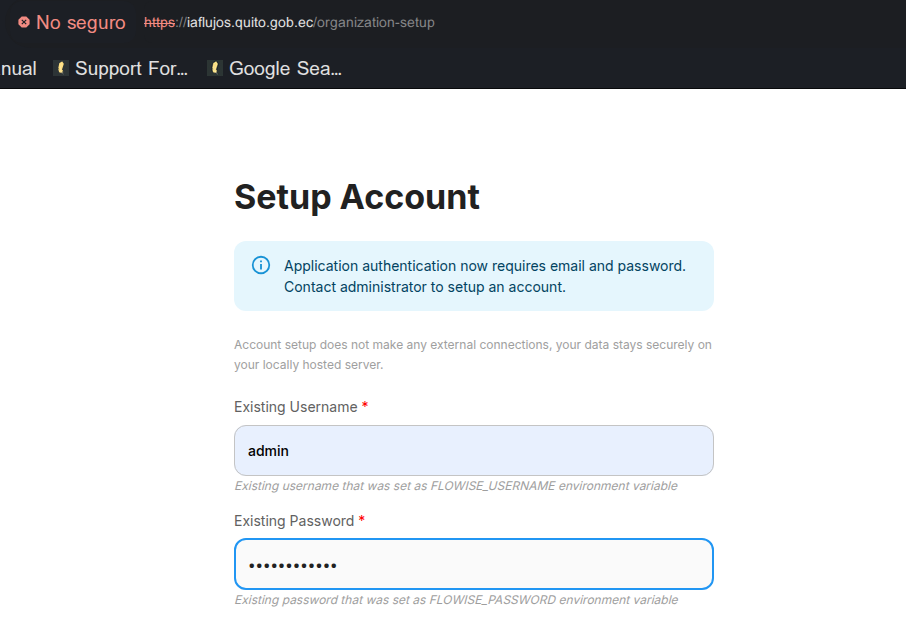
ping -c4 iaflujos.quito.gob.ec

4️⃣ Acceder a la aplicación y configurar - "esperar hasta que levanten bien los contenedores" : Aceptar el riesgo del certificado no seguro porque es autofirmado.

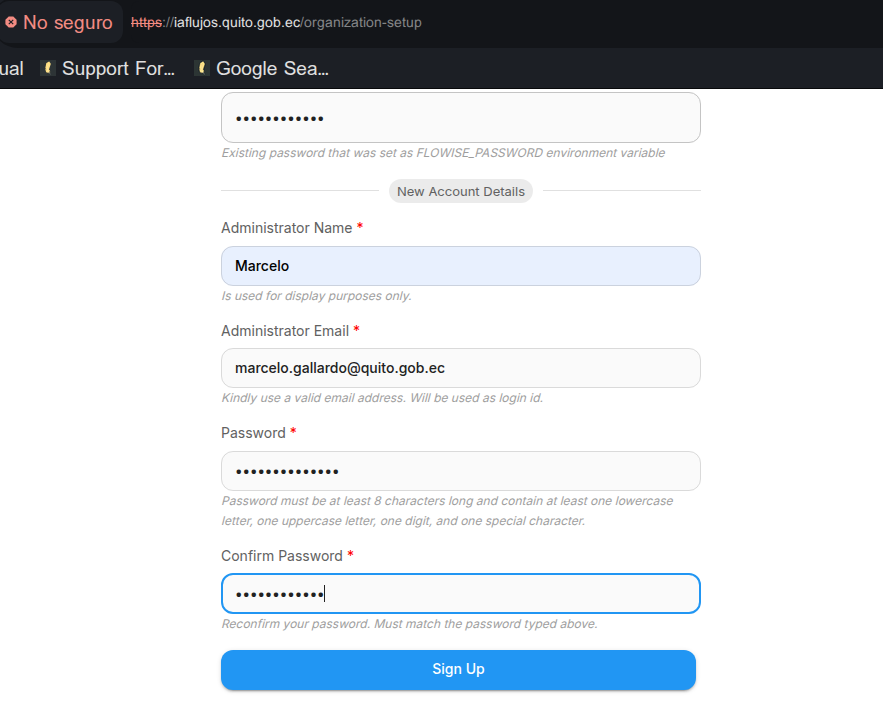
[https://iaflujos.quito.gob.ec](https://iaflujos.quito.gob.ec/)

🔑 Credenciales iniciales (solo para el primer arranque)

|  |  |
| --- | --- |
| **Usuario** | **Contraseña** |
| admin | Adm1n!202# |



📝 Registrar un usuario



📝 Ingresar a la administración de Flowise

