

## Docker Compose - Ecosistema de Servicios IoT

### Arquitectura de Orquestación de Contenedores

#### Descripción General

Este archivo docker-compose.yml define un ecosistema completo de servicios para una plataforma IoT, integrando componentes de gestión, bases de datos, brokers de mensajería, visualización y aplicaciones personalizadas. La configuración sigue una arquitectura de microservicios conectados a través de una red dedicada.

#### Servicios de Gestión e Infraestructura

##### Portainer - Gestión de Contenedores

Propósito: Interfaz web para administración y monitoreo de contenedores Docker

Imagen: portainer/portainer-ce:latest

Puerto: 9000

Volúmenes:

Socket de Docker para gestión

Datos persistentes de configuración

Función: Permite gestión visual del cluster de contenedores

##### Nginx Proxy Manager - Reverse Proxy y SSL

Propósito: Gestión centralizada de proxies reversos y certificados SSL

Imagen: jc21/nginx-proxy-manager:latest

Puertos:

80 (HTTP)

443 (HTTPS)

81 (Panel administrativo)

Configuración: Base de datos SQLite integrada

Función: Terminación SSL y enrutamiento de tráfico

##### Cloudflared - Túnel Cloudflare

Propósito: Exposición segura de servicios mediante túnel Cloudflare

Imagen: cloudflare/cloudflared:latest

Configuración: Token de autenticación embebido

Función: Conexión segura sin abrir puertos en el firewall

Servicios de Datos y Almacenamiento

MariaDB - Base de Datos Relacional

Propósito: Almacenamiento estructurado de datos de aplicación

Imagen: mariadb:latest

Variables: Credenciales root y usuario de aplicación

Volúmenes: Datos persistentes de MySQL

Función: Base de datos principal para la aplicación IoT

InfluxDB - Base de Datos de Series Temporales

Propósito: Almacenamiento optimizado para datos de sensores

Imagen: influxdb:1.8 (versión específica para compatibilidad)

Variables: Base de datos, usuario y contraseña administrativa

Función: Almacenamiento de métricas de sensores con alta frecuencia

Servicios de Mensajería y Procesamiento

Mosquitto MQTT - Broker de Mensajes

Propósito: Comunicación pub/sub para dispositivos IoT

Imagen: eclipse-mosquitto:2

Puertos:

1883 (MQTT standard)

9001 (WebSocket para clientes web)

Volúmenes: Configuración, datos y logs persistentes

Función: Central de mensajería para dispositivos IoT

Telegraf - Colector de Métricas

Propósito: Recolección y envío de métricas del sistema

Imagen: telegraf:latest

Configuración: Archivo de configuración montado

Dependencias: InfluxDB y Mosquitto

Función: Agente de recolección de datos

## Servicios de Visualización y Análisis

### Node-RED - Flujos de Trabajo IoT

Propósito: Orquestación visual de flujos de datos IoT

Imagen: nodered/node-red:latest

Puerto: 1880 (Interfaz de flujos)

Características: Proyectos habilitados

Función: Automatización y procesamiento de datos

### Grafana - Dashboards de Visualización

Propósito: Paneles de control y visualización de métricas

Imagen: grafana/grafana:latest

Configuración: Usuario y contraseña administrativa

Plugins: Paneles adicionales preinstalados

Función: Visualización de datos de sensores y métricas

### Adminer - Gestor de Bases de Datos

Propósito: Interfaz web para administración de bases de datos

Imagen: adminer

Función: Acceso visual a MariaDB e InfluxDB

## Aplicación IoT Personalizada

### Backend IoT - Servicio Principal

Propósito: API principal y servicios backend de la aplicación

Construcción: Dockerfile personalizado desde repositorio Git

Puerto: 3000 (para desarrollo)

Variables de Entorno:

Autenticación JWT

Configuración Google OAuth

Conexiones a bases de datos

Configuración MQTT

Listas de control de acceso

Health Check: Verificación automática de salud

Dependencias: MariaDB y Mosquitto

Frontend IoT - Interfaz de Usuario

Propósito: Aplicación web cliente para usuarios finales

Construcción: Dockerfile personalizado para frontend

Puerto: 5000 (servido por Nginx)

Dependencias: Backend IoT

Configuración de Red y Conectividad

Red Personalizada silo-network

Driver: bridge

Propósito: Aislamiento y comunicación entre servicios

Ventajas:

Comunicación segura entre contenedores

Aislamiento de la red host

Nombres de servicio como DNS

Características de Producción

Gestión de Estado

Restart Policies: unless-stopped para resiliencia

Volúmenes Persistentes: Datos críticos fuera de contenedores

Health Checks: Monitoreo automático de servicios

Seguridad

Usuario no-root: Servicios ejecutan con privilegios mínimos

Variables de Entorno: Credenciales externalizadas

Red aislada: Comunicación controlada entre servicios

Escalabilidad

Servicios independientes: Escalado selectivo posible

Dependencias explícitas: Orden de inicio controlado

Recursos compartidos: Bases de datos centralizadas

Flujo de Datos del Sistema

Dispositivos IoT → Mosquitto MQTT

Mosquitto → Telegraf/Node-RED/Backend IoT

Telegraf → InfluxDB (métricas del sistema)

Backend IoT → MariaDB (datos de aplicación)

InfluxDB/MariaDB → Grafana (visualización)

Frontend → Backend IoT (interfaz usuario)

Consideraciones de Despliegue

Variables de Entorno Requeridas

Credenciales de bases de datos (MySQL, InfluxDB, Grafana)

Secretos JWT para autenticación

Configuración Google OAuth

Listas de control de acceso por email

Volúmenes Persistentes

Estructura de directorios local para datos persistentes:

```
./
├── nginx-proxy-manager/
├── services/
│   ├── silo/
│   │   ├── mariadb/data
│   │   ├── influxdb/data
│   │   ├── mosquitto/{config,data,log}
│   │   ├── nodered/data
│   │   └── grafana/data
└── portainer/data
```

Esta configuración docker-compose proporciona una plataforma IoT completa y production-ready, con todos los componentes necesarios para ingesta, procesamiento, almacenamiento y visualización de datos de sensores.