



## Tarea evaluable 1

En esta primera tarea, quiero que despleguemos una pila MING en Docker (Mosquitto, Node-RED, InfluxDB, y Grafana) para captar el precio de Bitcoin (BTC) desde la API pública de CoinGecko, para no gastar todas las peticiones disponibles lo solicitaremos cada 10 minutos. Con esos datos vamos a construir un cuadro de mando en Grafana que tenga, al menos, los siguientes elementos:

- Precio actual (panel tipo Gauge o similar).
- Gráfica de la última hora (serie temporal).
- Texto / métrica con la tendencia de las últimas 24 h (subida/bajada y %).

### Objetivos de aprendizaje

- Desplegar un stack de observabilidad/IoT ligero (MING) con Docker.
- Diseñar un flujo de ingesta con Node-RED consumiendo una API REST.
- Modelar y persistir datos de series temporales en InfluxDB (mediciones, campos, tags).
- Construir un dashboard en Grafana con paneles KPI y series temporales.
- Aplicar buenas prácticas (parametrización, manejo de errores, documentación).

### Requisitos funcionales

1. Despliegue en Docker Compose con red interna y volúmenes persistentes para InfluxDB y Grafana.
2. Ingesta cada 10 minutos desde CoinGecko (Node-RED: inject → HTTP request → function → Influx out).
3. Esquema en InfluxDB:
  - a. measurement: crypto\_price
  - b. tags: symbol="BTC", currency (p. ej., EUR), source="coingecko"
  - c. fields: price (float), change\_24h (float, opcional), ts\_api (timestamp opcional)
  - d. InfluxDB v2: org, bucket y token gestionados por variables de entorno/secrets.
4. Cálculo de tendencia 24 h, con una de estas opciones (a elegir):
  - a. Calcular en Grafana: panel Stat que compare el valor actual con el valor de hace 24 h y muestre Δ% (override/transform).
  - b. Calcular en Node-RED y almacenar el %\_change\_24h como field adicional (si usáis el endpoint que ya devuelve cambio 24 h, basta con mapearlo).
5. Dashboard en Grafana con, al menos:
  - a. Panel Precio actual.
  - b. Panel Gráfica última hora (intervalo fijo 1h, auto-refresh razonable).
  - c. Panel Tendencia 24 h: texto/indicador con flecha ↑/↓ y porcentaje.



## Robustez mínima:

- Reintentos en caso de fallo ( $\text{HTTP} \neq 200$ ) y log del error en Node-RED.
- Respeto por la cadencia (10 minutos) sin ráfagas innecesarias.

## Documentación:

- README.md con cómo levantar el stack, variables usadas, endpoints y usuario/contraseña por defecto (si los hay), y capturas del dashboard.
- Documento con:
  - Explicación general de la implementación.
  - Capturas que muestren el funcionamiento de la pila.
  - Diagrama del flujo Node-RED.
  - Justificación del esquema Influx.
  - Justificación de decisiones tomadas para el dashboard y capturas del mismo.

## Entrega:

- Repositorio (o carpeta) con:
  - docker-compose.yml
  - Flows de Node-RED (flows.json)
  - Ficheros .env/secrets (sin credenciales sensibles reales)
  - README.md
- PDF con el documento.

## Valoración (10 puntos)

- Infraestructura Docker (Compose, redes, volúmenes, parametrización) – 2 pts
- Flujo Node-RED – 2,5 pts
- Modelo de datos en InfluxDB y consultas – 2 pts
- Dashboard Grafana – 2,5 pts
- Documentación y entregables – 1 pto
- Extra (hasta +1 pto):
  - Alertas en Grafana por umbral de variación ( $>\pm 2\%$  en 1 h).
  - Variables de entorno y .env bien estructurado + secrets Docker.
  - Panel comparativo BTC/EUR vs BTC/USD o media móvil 1h.