

Proyecto #3 Interfaces Gráficas - INFO1128

By Alberto Caro – Ayudante Fernando Valdés

1.- La empresa 'Estoy Frito Ltda' lo ha contratado a usted para que desarrolle un prototipo de un **DashBoard** ambiental para analizar su potencial y ver si usted posee las habilidades presionales que se necesitan para liderar este proyecto a escala nacional. Para realizar este proyecto usted deberá investigar e instalar los siguientes recursos:

- 1.- **NodeRed** (<https://nodered.org/>)
- 2.- **Grafana** (<https://grafana.com/>)
- 3.- **Mosquitto Broker** (<https://mosquitto.org/>)
- 4.- Librería de **Python** → **paho-mqtt**

Deberá realizar lo siguiente:

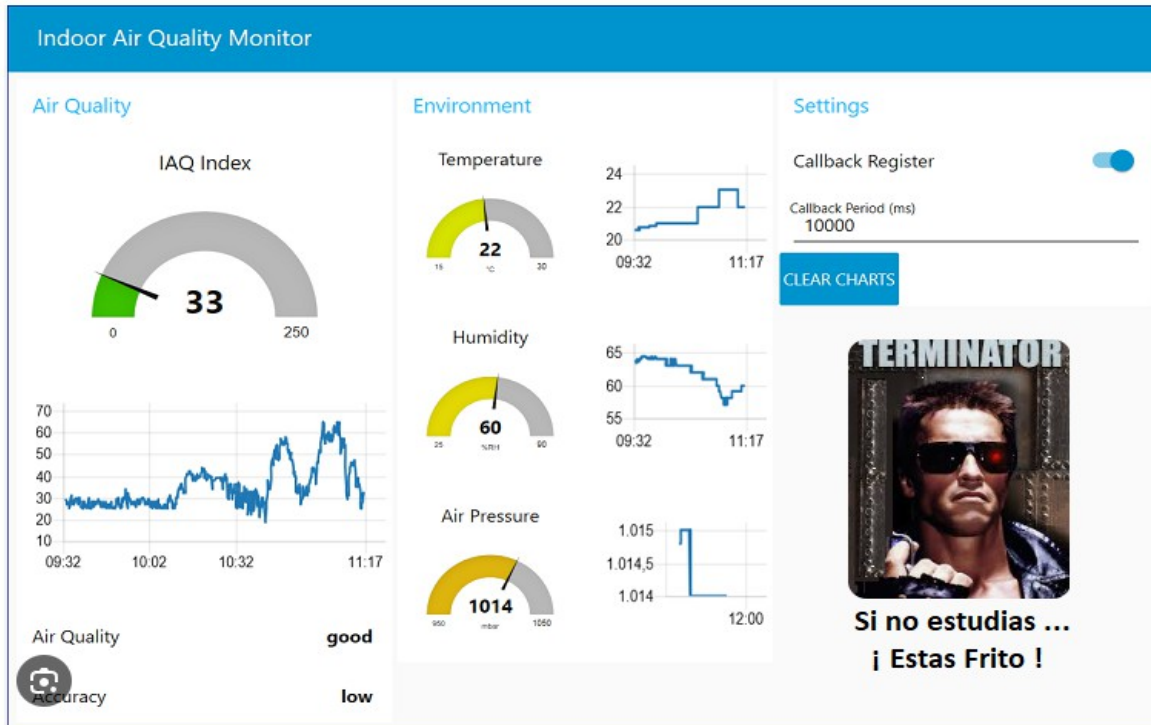
1. Desarrollar un script en **Python 3.x** que genere el siguiente **JSON**:

```
dUMA = {
    'te' : np.random.normal(20,2,1),      # Temperatura
    'hr' : np.random.normal(70,2,1),      # Humedad aire
    'pa' : np.random.normal(900,10,1),    # Presión atmosférica
    'p01' : np.random.normal(20,2,1),     # MP 1.0 ug/m3
    'p25' : np.random.normal(30,2,1),     # MP 2.5 ug/m3
    'p10' : np.random.normal(30,2,1),     # MP 10 ug/m3
    'h03' : np.random.normal(1000,10,1),  # Histograma MP 0.3 um
    'h05' : np.random.normal(1000,10,1),  # Histograma MP 0.5 um
    'h01' : np.random.normal(1000,10,1),  # Histograma MP 1.0 um
    'h25' : np.random.normal(1000,10,1),  # Histograma MP 2.5 um
    'h50' : np.random.normal(1000,10,1),  # Histograma MP 5.0 um
    'h10' : np.random.normal(1000,10,1)  # Histograma MP 10 um
}
```

2. El **JSON** anterior se debe enviar por **Python 3.x HTTP POST** a un **DashBoard** en **NodeRed**.
3. El **JSON** anterior se debe enviar por **Python 3.x HTTP POST** a un **DashBoard** en **Grafana**.
4. El **JSON** anterior se debe enviar por **Python 3.x** mediante protocolo **MQTT** a un **DashBoard** en **NodeRed**. Debe tener corriendo **Mosquitto Broker**. Aquí debe programar un script en **Python 3.x** que **Publique** los datos **JSON** en el Tópico '**sensores**' y un script **Python 3.x Suscriptor** que envíe este **JSON** a un **DashBoard** en **NodeRed**. Configure **Mosquitto Broker** para que la conexión sea segura con **password** y **clave** a su elección para los **Publicadores** y **Suscriptores**.

Se muestran algunas interfaces en **NodeRed** y **Grafana** a modo de ejemplo las cuales usted deberá

adaptar para que visualicen todos los datos del **JSON** anterior.



NodeRed



Grafana

El objetivo de este proyecto es que usted sea capaz de conocer nuevas tecnologías que serán utilizadas para desplegar información numérica y gráfica en el contexto de interfaces gráficas. Investigue e instale todos los recursos anteriores y sea creativo en el diseño de los **DashBoard** donde se deberá mostrar toda la información del **JSON** que envía **Python** de manera aleatoria.

Observaciones

1.- Si el proyecto se realiza en **Grupo de 3** personas:

- Una persona es responsable → **NodeRed**
- Una persona es responsable → **Grafana**
- Una persona es responsable → **Mosquitto Broker**
- Si es individual → **Responsable de todo.**

2.- Fecha de presentación y defensa: Jueves **20** en oficina del profesor, desde las **09:00 – 17:00**.

Rúbrica de Evaluación Simplificada: Proyecto #3 Interfaces Gráficas

Puntaje Total: 100 puntos

(Nota: 60 puntos obtenidos en esta rúbrica equivalen a un 4.0 en la escala de 1 a 7 puntos)

Categoría de Evaluación	Puntos Máximos	Criterios Clave de Calificación
1. Configuración de Herramientas	10	Instalación y funcionamiento correcto de NodeRed, Grafana, Mosquitto y la librería `paho-mqtt` de Python.
2. Generación de JSON (Python)	15	Script Python que crea el objeto JSON `dUMA` con la estructura y datos aleatorios especificados.
3. Envío de Datos HTTP POST	30	El JSON se envía correctamente desde Python a NodeRed y a Grafana vía HTTP POST, siendo recibido y procesado en ambos.
4. Flujo de Datos MQTT	30	Implementación completa del protocolo MQTT: Mosquitto Broker seguro, script publicador, script suscriptor, y recepción en NodeRed.
5. Dashboards y Visualización	15	Los Dashboards en NodeRed y Grafana presentan todos los datos del JSON de forma clara, funcional y creativa.
Total	100	

