

# CONESCAPANHONDURAS2025paper122.pdf

 Institute of Electrical and Electronics Engineers (IEEE)

---

## Document Details

### Submission ID

trn:oid:::14348:477757220

### Submission Date

Jul 31, 2025, 10:23 PM CST

### Download Date

Aug 12, 2025, 6:26 PM CST

### File Name

CONESCAPANHONDURAS2025paper122.pdf

### File Size

369.6 KB

4 Pages




2,331 Words

13,388 Characters

# 12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Top Sources

- 12%  Internet sources
- 5%  Publications
- 0%  Submitted works (Student Papers)

## Integrity Flags




### 0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Top Sources

- 12%  Internet sources
- 5%  Publications
- 0%  Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	arxiv.org	5%
2	Internet	opus.hs-furtwangen.de	2%
3	Internet	repositorioacademico.upc.edu.pe	2%
4	Internet	db-engines.com	<1%
5	Internet	www.amazon.in	<1%
6	Publication	Silvia Marzal Romeu. "Concepción e integración de arquitecturas y protocolos de ...	<1%
7	Internet	bermellelectromedicina.com	<1%
8	Internet	www.coursehero.com	<1%
9	Internet	www.enfermeriaencardiologia.com	<1%
10	Internet	www.randstad.com	<1%
11	Internet	ir.lib.uth.gr	<1%

# Integration of Grafana and MQTT for Near-Space Sensor Data Visualization: A Case Study with the EOHAB-1 Platform

1

1<sup>st</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

2<sup>nd</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

3<sup>rd</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

4<sup>th</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

5<sup>th</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

6<sup>th</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

**Abstract**—Monitoring near-space sondes is crucial for collecting environmental data under extreme conditions. This paper presents the proposed integration of Grafana and MQTT for real-time visualization of sensor data collected by the EOHAB1 platform. A communication pipeline using LoRa modules, local processing scripts, and database storage was developed. Preliminary tests using simulated data validate the functionality of the dashboard for mission planning and telemetry monitoring.

**Index Terms**—Grafana, MQTT, LoRa, dashboard, telemetry monitoring, HAB

## I. INTRODUCCIÓN

Este artículo presenta la integración de Grafana, una herramienta de código abierto para visualización de datos, con el protocolo de mensajería ligera MQTT (Message Queuing Telemetry Transport). La combinación de estas tecnologías permite desplegar dashboards en tiempo real que presentan los datos conforme son recibidos desde la sonda por la estación base.

La arquitectura propuesta utiliza módulos LoRa como medio de transmisión de largo alcance y bajo consumo energético. Los datos recibidos por la estación terrestre son procesados localmente y publicados en un broker MQTT, desde donde son consumidos por una capa de almacenamiento (como InfluxDB o SQL Server), habilitando su representación gráfica a través de Grafana.

El sistema ha sido diseñado con un enfoque modular, permitiendo futuras ampliaciones como geolocalización dinámica, integración con interfaces de control de misión, o incorporación de mecanismos de alerta. Finalmente, se describe la arquitectura de hardware, los protocolos de comunicación, la canalización de datos basada en MQTT y se

presenta un caso de estudio con datos simulados que valida el funcionamiento del sistema en condiciones de laboratorio.

## II. ARQUITECTURA DEL SISTEMA

### A. Descripción general del EOHAB-1

El EOHAB-1 es una plataforma de observación atmosférica basada en un globo de gran altitud (High Altitude Balloon, HAB), diseñada para misiones científicas en la estratósfera. Esta sonda transporta un conjunto de sensores integrados que permiten monitorear variables ambientales críticas durante el ascenso, flotación y descenso del globo. Entre los sensores incorporados se encuentran:

- Sensor de temperatura: para medir las variaciones térmicas en diferentes capas de la atmósfera.
- Sensor de presión barométrica: fundamental para estimar la altitud y estudiar el comportamiento atmosférico.
- Unidad de Medición Inercial (IMU): que proporciona datos de aceleración y orientación, útiles para analizar la dinámica del vuelo.
- Módulo GPS: encargado de registrar la posición geográfica, velocidad y trayectoria del globo en tiempo real.

Toda la información capturada se transmite mediante tecnologías de comunicación de largo alcance hacia una estación terrestre, donde es procesada, almacenada y visualizada para su análisis en tiempo real y posterior estudio.

### B. Arquitectura del hardware

La arquitectura de hardware del sistema EOHAB-1 está compuesta por varios subsistemas que interactúan para adquirir, transmitir y soportar el flujo de datos desde el globo hasta la estación terrestre.

3

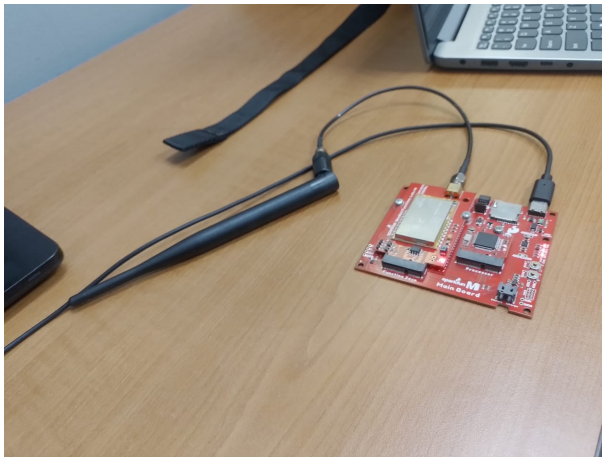


Fig. 1. Micromod LoRa y antena

1) *Plataforma de adquisición a bordo (EOHAB-1)*: El núcleo del sistema a bordo consiste en un microcontrolador de bajo consumo que coordina la lectura de múltiples sensores ambientales y de navegación, incluyendo los anteriormente mencionados. Todos estos sensores están conectados al microcontrolador a través de buses I2C y son consultados periódicamente para generar un paquete de datos estructurado.

2) *Módulo de transmisión LoRa*: Una vez formateados los datos, estos son enviados a través de un módulo LoRa MicroMod operando a 915 MHz. Este módulo ha sido seleccionado por su capacidad de transmisión de largo alcance con bajo consumo energético, lo que resulta ideal para misiones estratosféricas. La antena utilizada es de tipo omnidireccional o helicoidal, optimizada para comunicaciones verticales en línea de vista.

3) *Estación terrestre receptora*: En tierra, se dispone de una estación de recepción compuesta por: un módulo LoRa receptor del mismo tipo que el emisor, una antena direccional o de alta ganancia, ajustada manualmente para maximizar la señal entrante y una computadora de procesamiento conectada vía USB-Serial al receptor LoRa, la conexión de los elementos se aprecia en la Fig. 1.

### III. ESTRUCTURA DEL PAQUETE DE DATOS

El envío de datos desde el sistema EOHAB-1 a la estación terrestre requiere una estructura de paquete eficiente, compacta y tolerante a errores. Para lograrlo, se ha diseñado un formato de paquete personalizado que permite encapsular los datos más relevantes en un mensaje de longitud controlada, garantizando integridad y facilidad de procesamiento. La estructura del paquete de datos se muestra en la Fig. 2.

El encabezado define la estructura básica del paquete y puede contener:

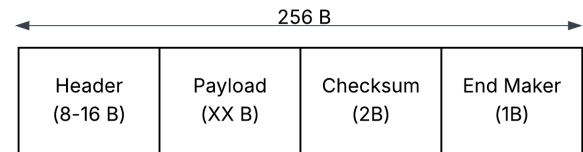


Fig. 2. Estructura del paquete de datos

- Tipo de datos: Identificador del subsistema (ADCS, EPS, Payload).
- ID del paquete: Número único para detectar paquetes duplicados.
- Tamaño del payload: Bytes de datos útiles en el paquete.
- Flags: Opciones como prioridad, retransmisión, etc.

Datos fundamentales: Lat, Lot, Alt, Timestamp, Velocidad en Z, Estado de carga %, STRING de Status de Payload, Mode Status Char.

### IV. IMPLEMENTACIÓN DE GRAFANA PARA LA VISUALIZACIÓN DE LOS DATOS

El monitoreo de sensores y sistemas de control a través de dashboards representa una solución eficiente, compacta e interactiva para la visualización de datos en misiones near-space. Los datos transmitidos por los distintos subsistemas del satélite y recibidos por la estación terrestre deben ser organizados de manera estructurada y clara, maximizando el uso del espacio visual disponible para facilitar la interpretación en tiempo real. Esto es fundamental para garantizar el seguimiento adecuado del estado operacional de los sensores y del sistema en su conjunto.

Grafana es una plataforma de código abierto ampliamente utilizada para la visualización de datos métricos en tiempo real [1], [2]. Ofrece soporte para múltiples fuentes de datos, incluyendo bases de datos de series temporales como InfluxDB, Graphite y OpenTSDB, así como también bases de datos relacionales como SQL Server y MySQL [3]. Grafana permite construir paneles de control altamente personalizables que integran gráficos, alertas y consultas dinámicas, lo que lo convierte en una herramienta ideal para entornos de monitoreo continuo y análisis técnico en tiempo real [4].

Grafana funciona de forma efectiva con conexiones a bases de datos como SQL Server; sin embargo, requiere una configuración adecuada para garantizar la comunicación eficiente entre la capa de datos y la visualización. En entornos IoT, Grafana es frecuentemente integrado junto a herramientas como Node-RED y Telegraf, ampliando así sus capacidades para el tratamiento y análisis de flujos de datos provenientes de sensores distribuidos [5].

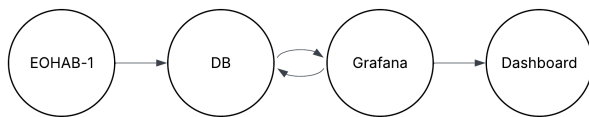


Fig. 3. Diagrama de flujo de datos general

	date	tiempo	latitude	longitude	altitude_MSL	verticalSpeed	OutsideTemperature_Kelvin	OutsidePressure_PA	TotalReistros
1	4/23/2024	10:56:32 AM	13.81	-89.33	504.00	0.00	284.87	95415.97	9020
2	4/23/2024	10:56:33 AM	13.81	-89.33	505.01	1.90	284.87	95404.38	9020
3	4/23/2024	10:56:34 AM	13.81	-89.33	507.46	2.85	284.85	95376.34	9020
4	4/23/2024	10:56:35 AM	13.81	-89.33	510.50	3.17	284.83	95341.64	9020
5	4/23/2024	10:56:36 AM	13.81	-89.33	513.72	3.26	284.81	95304.81	9020
6	4/23/2024	10:56:37 AM	13.81	-89.33	516.99	3.28	284.79	95267.43	9020
7	4/23/2024	10:56:38 AM	13.81	-89.33	520.28	3.29	284.77	95229.90	9020
8	4/23/2024	10:56:39 AM	13.81	-89.33	523.57	3.29	284.75	95192.34	9020
9	4/23/2024	10:56:40 AM	13.81	-89.33	526.86	3.29	284.73	95154.77	9020
10	4/23/2024	10:56:41 AM	13.81	-89.33	530.15	3.29	284.70	95117.21	9020

Fig. 4. Base de datos de sensores simulada

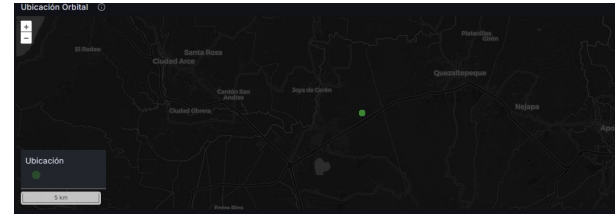


Fig. 5. Geomapa en Grafana

## A. Protocolo de transmisión de datos en tiempo real

Con el objetivo de habilitar la visualización en tiempo real de los datos recolectados por la sonda near-space, se ha implementado una arquitectura de comunicación basada en el protocolo ligero MQTT. Este protocolo, ampliamente adoptado en entornos de IoT, se adapta especialmente bien a sistemas con recursos limitados y enlaces de comunicación intermitentes, como los que caracterizan a misiones estratosféricas.

El modelo publish-subscribe de MQTT permite desacoplar los dispositivos emisores (publishers) de los receptores (subscribers), facilitando una canalización de datos flexible y escalable entre la estación receptora y la plataforma de visualización. En este sistema, los datos recibidos por el módulo LoRa en la estación terrestre son procesados por una aplicación local que actúa como cliente MQTT publisher, encargada de estructurar los datos y enviarlos a un broker MQTT, que puede estar alojado localmente o en la nube.

El flujo de datos completo puede describirse de la siguiente manera:

- 1) Módulo LoRa (Receptor): recibe en tiempo real los paquetes transmitidos por la sonda EOHAB1.
- 2) Script de procesamiento local: deserializa, valida e interpreta los datos entrantes, estructurándolos en un formato estandarizado (por ejemplo, JSON).
- 3) Cliente MQTT: publica los datos procesados en topics específicos dentro del broker MQTT.
- 4) Broker MQTT: intermedia los mensajes y los distribuye a todos los clientes suscritos.
- 5) Servicio de almacenamiento intermedio: herramientas como Telegraf o Node-RED se suscriben a los topics, transforman los datos según sea necesario (agregación, conversión de unidades, etiquetado temporal) y los almacenan en una base de datos como InfluxDB o SQL Server.
- 6) Grafana: consulta directamente la base de datos para generar dashboards personalizados que visualizan los datos en tiempo real mediante gráficos, indicadores y mapas.

Este flujo de comunicación proporciona una arquitectura robusta y modular para el monitoreo remoto de misiones near-space. No obstante, durante las etapas iniciales de desarrollo y validación del sistema, se ha utilizado una versión simplificada

de esta arquitectura basada en registros simulados, tal como se aprecia en la Fig. 3.

## B. Construcción del dashboard

La construcción de un dashboard en Grafana se hace mediante consultas TransactSQL que traen los datos que el usuario pide y en base a estas consultas pueden visualizarlas con distintos gráficos, depende de lo que se necesite y los datos que se tengan.

Para las primeras pruebas se hizo uso de la base de datos de la Fig. 4, denominada dummyData (datos de prueba con sensores), la cual contiene data de prueba aprovechable para realizar consultas y visualizaciones necesarias para una primera prueba del dashboard. Esta Database de más de 9,000 registros contiene una tabla con los campos de:

- 1) Altitud
- 2) Longitud
- 3) Latitud
- 4) Tiempo (hh:mm:ss)
- 5) Fecha
- 6) Presión atmosférica
- 7) Velocidad
- 8) Temperatura

Un dashboard en Grafana se compone de paneles, en ellos se implementan consultas SQL y se manipulan mediante a opciones de gráficos y parámetros ya establecidos en Grafana y a su vez del tipo de data que se quiere mostrar. En la Fig.5. se muestra un ejemplo con la data de ubicación y su gráfico Geomapa de Grafana.

La visualización, como se ha planteado anteriormente, requiere de data correcta, consulta correcta y grafico correcto; esto asegura una vista de panel asertivo, además, unos parámetros bien establecidos ayudan a mejorar y entender mejor los gráficos.



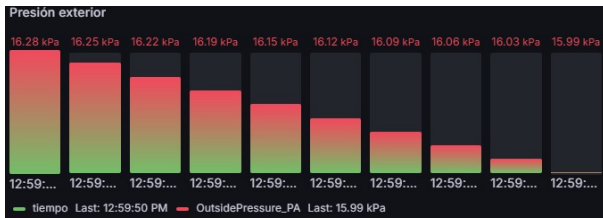


Fig. 6. Gráfico de líneas de presión atmosférica en Grafana

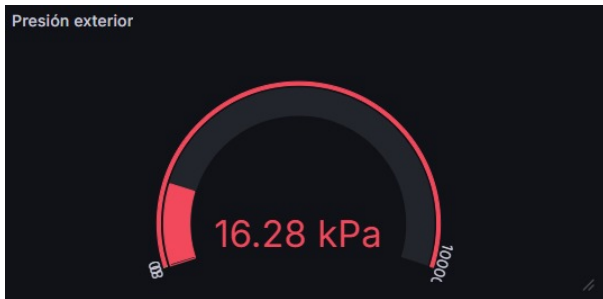


Fig. 7. Indicador de presión atmosférica en Grafana

## V. RESULTADOS

En la etapa actual del proyecto, se ha logrado construir un entorno de visualización funcional utilizando Grafana y una base de datos de pruebas, con el objetivo de validar la arquitectura propuesta y evaluar su viabilidad para la misión EOHAB1. Si bien la integración completa del protocolo MQTT aún no ha sido implementada en condiciones de vuelo, se han realizado pruebas controladas en laboratorio utilizando una base de datos denominada dummyData, la cual simula los valores típicos de sensores embarcados en la sonda near-space.

Esta base de datos contiene más de 9,000 registros de variables como altitud, latitud, longitud, temperatura, presión atmosférica y velocidad vertical, generadas con patrones realistas para validar las capacidades de consulta y visualización en tiempo real. Los dashboards en Grafana fueron construidos empleando paneles interactivos y consultas Transact-SQL, permitiendo explorar el comportamiento de los datos en distintos formatos gráficos, incluyendo:

- Geomaps basados en coordenadas GPS, como el de la Fig. 5.
- Gráficos de líneas para tendencias de temperatura, presión y altitud, como el de la Fig. 6.
- Indicadores instantáneos con rangos configurables, como se muestra en la Fig. 7.

Los resultados obtenidos demuestran que la arquitectura propuesta puede manejar adecuadamente el volumen y tipo de datos esperado en una misión near-space. La modularidad del sistema permite adaptar el backend de datos (InfluxDB o SQL Server) sin alterar la lógica de visualización, y la

construcción de paneles en Grafana es lo suficientemente flexible para representar distintos subsistemas o condiciones de vuelo.

En fases posteriores, se contempla la incorporación del cliente MQTT publisher en el script de procesamiento local, así como la implementación de un broker MQTT y la configuración de un servicio como Telegraf o Node-RED para gestionar el almacenamiento dinámico de datos en InfluxDB.

La validación de la arquitectura en esta primera etapa permite establecer una base técnica sólida sobre la cual desplegar las pruebas con datos en tiempo real, tanto en vuelos de prueba como en campañas de lanzamiento de la sonda.

## VI. CONCLUSIONES

Este trabajo presenta el diseño e implementación parcial de una arquitectura de visualización de datos para misiones near-space, utilizando herramientas libres y eficientes como Grafana para la visualización y MQTT como protocolo de comunicación propuesto. A pesar de que la integración completa con MQTT aún no ha sido implementada, se ha logrado validar satisfactoriamente la estructura de almacenamiento y visualización mediante el uso de datos simulados representativos del entorno operativo de la plataforma EOHAB1-1.

Las pruebas realizadas con dashboards en Grafana y datos almacenados en una base SQL han demostrado que el sistema propuesto es funcional, flexible y capaz de representar variables clave como altitud, temperatura, presión y geolocalización de forma efectiva e interactiva. Esto permite anticipar una integración exitosa con flujos de datos reales en futuras etapas del proyecto.

## REFERENCES

- [1] G. Boros, "Visualizing IoT Data with MQTT, QuestDB, and Grafana," \*Medium\*, Mar. 2023. [Online]. Available: <https://gabor-boros.medium.com/visualizing-iot-data-with-mqtt-questdb-and-grafana-9d773af3673e>
- [2] HiveMQ, "MQTT Data Visualization with Grafana," \*HiveMQ Blog\*, Oct. 2022. [Online]. Available: <https://www.hivemq.com/blog/mqtt-data-visualization-with-grafana/>
- [3] Grafana Labs, "Streaming real-time sensor data to Grafana using MQTT and Grafana Live," \*Grafana Labs Documentation\*, 2021. [Online]. Available: <https://grafana.com/blog/2021/08/12/streaming-real-time-sensor-data-to-grafana-using-mqtt-and-grafana-live/>
- [4] Apptimia, "Streaming Real-Time IoT Sensor Data to Grafana with MQTT," \*Apptimia Blog\*, 2023. [Online]. Available: <https://www.apptimia.com/blog/streaming-real-time-iot-sensor-data-to-grafana-with-mqtt/>
- [5] D. Kairuz Cabrera et al., "Integration of Node-RED and Grafana in a Weather Monitoring System," in \*Proc. IEEE ANDESCON\*, 2020.