Boletim da Sociedade Astronômica Brasileira, **37**, no. 1, 51-52

Modular Telemetry Infrastructure for Astronomical Observatories

An Application at the Pico dos Dias Observatory.

B. Antoniolli[1,2], L. Fraga[2], & M. Zanetti[1]

[1] Universidade Federal de Santa Maria, Santa Maria, RS, Brasil, e-mail: bruna.antoniolli@acad.ufsm.br
[2] Laboratório Nacional de Astrofísica, Itajubá, MG, Brasil,e-mail: bantoniolli@lna.br,lfraga@lna.br

Abstract. This work presents the development of a modular infrastructure for the acquisition, routing, storage, and visualization of real-time telemetry data in astronomical observatories. The proposal utilizes widely adopted open-source technologies for modern distributed systems, such as the messaging protocols MQTT and ZeroMQ, the time-series database InfluxDB, and interactive dashboards and alert configuration with Grafana. The designed architecture is oriented towards the integration of multiple technical and scientific subsystems present in astronomical observatories, including telescopes, detectors, environmental sensors, and weather stations, among others; by means of an observable and scalable mesh. As a case study, the platform will be applied to the Pico dos Dias Observatory (OPD), contributing to the modernization of its infrastructure and expanding the capacity for remote diagnosis, predictive maintenance, and operational response. The project involves the incremental development of components in Python, testing with real and simulated sensors, and the integration of automatic alerts, configured in a virtualized environment with Docker. The expected final product is a functional and reproducible solution that enhances operational reliability and facilitates remote diagnoses and predictive maintenance at the OPD, applicable to other small and medium-sized observatories.

Resumo. Este trabalho apresenta o desenvolvimento de uma infraestrutura modular para aquisição, roteamento, armazenamento e visualização de dados de telemetria em tempo real em observatórios astronômicos. A proposta utiliza tecnologias de código aberto amplamente adotadas em sistemas distribuídos modernos, como os protocolos de mensageria MQTT e ZeroMQ, banco de dados de séries temporais InfluxDB, e dashboards interativos e configuração de alertas com Grafana. A arquitetura projetada é orientada à integração de múltiplos subsistemas técnicos e científicos presentes em observatórios astronômicos, incluindo telescópios, detectores, sensores ambientais e estação meteorológica, entre outros; por meio de uma malha observável e escalável. Como estudo de caso, a plataforma será aplicada ao Observatório do Pico dos Dias (OPD), contribuindo para a modernização de sua infraestrutura e ampliando a capacidade de diagnóstico remoto, manutenção preditiva e resposta operacional. O projeto envolve o desenvolvimento incremental de componentes em Python, testes com sensores reais e simulados, e integração de alertas automáticos, configurados em ambiente virtualizado com Docker. Espera-se como produto final uma solução funcional e reprodutível que aprimora a confiabilidade operacional e facilita diagnósticos remotos e manutenção preditiva no OPD, aplicável a outros observatórios de pequeno e médio porte.

Keywords. Astronomical observatories

1. Introduction

Modern observatories require robust telemetry systems to support increasingly automated and robotic operations. The National Astrophysics Laboratory (LNA), through the Pico dos Dias Observatory (OPD), is modernizing its infrastructure, creating a critical need to integrate diverse data sources such as telescopes, environmental sensors, and dome controls.

The objective of this work is to design and unified, scalable, and open-source telemetry platform. Built on a modular architecture, the system manages data acquisition, routing, storage, and visualization. This infrastructure is intended to support distributed operations, enabling efficient remote diagnostics and increasing operational reliability with low maintenance costs.

2. Architecture

The proposed architecture is developed within a controlled test environment, leveraging Docker container technology to ensure portability, service isolation, and system reproducibility, while minimizing dependence on the host machine configuration. The methodology adopts a practical and incremental approach, focusing on the integration of both real and simulated sensors. This strategy allows for continuous system validation through field observations conducted directly at the OPD.

Telemetry data collected from the observatory's various subsystems are stored in InfluxDB, a database optimized for time-series data, enabling efficient queries with high temporal resolution. Data visualization is provided by interactive dashboards developed in Grafana, which support real-time monitoring of subsystem operational status and the generation of automatic alerts for critical events.

The system will be structured as a modular software architecture composed of independent units dedicated to:

- Data Acquisition: Communication with sensors and devices via modern protocols such as MQTT and ZeroMQ;
- Message Routing and Processing: Ensuring reliability and flexibility in data transmission;
- Structured Storage: Persistence in a time-series database (InfluxDB);
- Interactive Visualization: Operational dashboards developed in Grafana;
- Critical Event Notification: Automated alert channels for anomaly detection.

Figure 1 illustrates the flowchart of the baseline architecture.

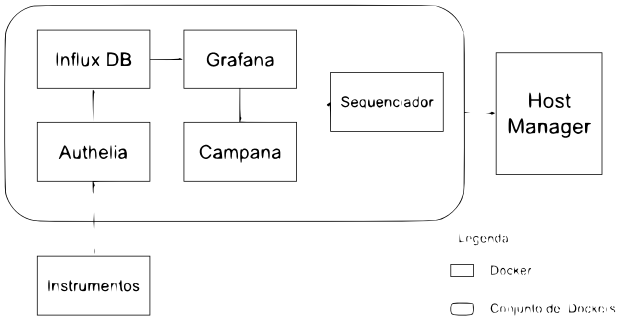

Figure 1. Flowchart of the proposed data architecture. The pipeline captures data from Instrumentos (sensors), secured by Authelia, into InfluxDB (storage). Grafana visualizes states, triggering Campana (alerts) and the Sequenciador (orchestration) to update the Host Manager.

3. Methodology

The infrastructure development followed an incremental approach, progressing from simulation to containerization and final integration.

3.1. Telemetry Simulation

To validate the architecture without disrupting observatory operations, we developed Python-based simulators. Adhering to ASCOM standards, these scripts act as virtual devices, generating JSON payloads. This aproach allowed us to stress-testing the message brokers and database ingestion pipeline prior to physical deployment.

3.2. Containerization and Orchestration

The system relies on Docker for reproducibility. A `docker-compose` environment orchestrates the interaction between four key services: an MQTT Broker for messaging; an InfluxDB container for time-series storage; a custom Python Ingester that subscribes to topics and writes to the database; and Grafana for dashboard visualization.

3.3. Pipeline Integration

The data pipeline operates on a hub-and-spoke topology. Simulators publish telemetry to specific MQTT topics (e.g., `opd/telescope`). The Ingester service converts these payloads into InfluxDB Line Protocol. Finally, Grafana utilizes the Flux query language to fetch this data for real-time visualization and alert evaluation.

4. Initial Results

Validation in the simulation environment confirmed the stack's stability and scalability. The automated provisioning enabled loss-free ingestion of high-frequency telemetry, maintaining performance even after the seamless integration of a secondary simulator into the data pipeline.

The operational visualization capability is highlighted in Fig. 2. The system applies real-time conditional logic to the dome azimuth, automatically flagging threshold violations (> 80) in red for immediate anomaly detection. These results attest to the platform's reliability and readiness for monitoring critical observatory subsystems.

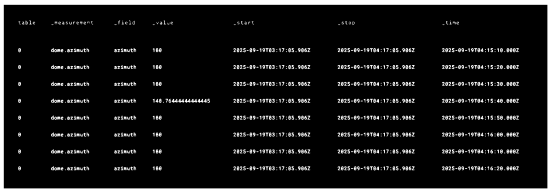

Figure 2. Data persistence verification in InfluxDB. The figure displays the raw time-series structure for the 'dome.azimuth' measurement. Each record is indexed by a precision timestamp, confirming the successful ingestion and structured storage of sensor telemetry for historical analysis.

5. Conclusions

This work presents the design and initial validation of a functional and reproducible telemetry infrastructure for the Pico dos Dias Observatory (OPD), built using Docker containers and open-source technologies to ensure portability and scalability. Validation in a simulated environment has confirmed the stability of end-to-end data pipeline, which spans MQTT, InfluxDB, and Grafana and has demonstrated the ability to perform real-time anomaly detection through conditional alerting. The proposed modular architecture offers a cost-effective and adaptable solution for diverse astronomical subsystems, and can be extended to other smal and medium-sized observatories. Future work on-site deployment at the OPD, integration with physical sensors, and the implementation of predictive maintenance strategies to enhance operational reliability.

Bibliography

Fausti Neto, A. et al. 2024, Sasquatch: Rubin Observatory metrics and telemetry service. In: *SPIE Astronomical Telescopes + Instrumentation*, Online Only. Proceedings of SPIE, v. 13101, p. 131011M.

Huertas, M. et al. 2016, Monitoring Service for the Gran Telescopio Canarias Control System. In: *SPIE Astronomical Telescopes + Instrumentation*, Edinburgh, United Kingdom. Proceedings of SPIE, v. 9913, p. 99133P.

Koopman, B. J. et al. 2024, The Simons Observatory: Deployment of the observatory control system and supporting infrastructure. In: *SPIE Astronomical Telescopes + Instrumentation*, Online Only. Proceedings of SPIE, v. 13101, p. 131014D.

van Gend, C. et al. 2024, The Intelligent Observatory: software for robotic operation, monitoring and safeguarding. In: *SPIE Astronomical Telescopes + Instrumentation*, Online Only. Proceedings of SPIE, v. 13101, p. 131012W.