



UFSM

LNA



**UNIVERSIDADE FEDERAL DE SANTA MARIA - UFSM
LABORATÓRIO NACIONAL DE ASTROFÍSICA - LNA**

PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA - PIBIC

**Infraestrutura de Telemetria, Monitoramento e Logging
para o Observatório do Pico dos Dias
Relatório Final**

Aluno

Bruna Eduarda Antonioli
Engenharia Aeroespacial

Orientador

Luciano Fraga
Laboratório Nacional de Astrofísica

Marcelo Serrano Zanetti
Universidade Federal de Santa Maria

PIBIC/CNPq
Abril/2025 a Fevereiro/2026

**Santa Maria, RS, Brasil
Fevereiro - 2026**

Resumo

O presente trabalho propõe o desenvolvimento de uma infraestrutura modular para aquisição, armazenamento e visualização de dados de telemetria em tempo real em observatórios astronômicos. A solução utiliza tecnologias de código aberto, como MQTT, ZeroMQ, InfluxDB e Grafana, visando integrar diferentes subsistemas, incluindo telescópios, detectores e sensores ambientais, em uma arquitetura observável e escalável. A plataforma é aplicada ao Observatório do Pico dos Dias (OPD), contribuindo para a modernização de sua infraestrutura e ampliando a capacidade de monitoramento, diagnóstico remoto e manutenção preditiva. O projeto envolve o desenvolvimento de componentes em Python, integração de alertas automáticos e virtualização com Docker, resultando em uma solução funcional e, acessível e reprodutível aplicável a outros observatórios de pequeno e médio porte

Palavras-chave: Telemetria, Mensageria, Observatórios Astronômicos, InfluxDB, MQTT, Grafana, Modernização.

Abstract

This work proposes the development of a modular infrastructure for the acquisition, storage, and visualization of real-time telemetry data in astronomical observatories. The solution employs widely adopted open-source technologies, such as MQTT, ZeroMQ, InfluxDB, and Grafana, aiming to integrate different subsystems — including telescopes, detectors, and environmental sensors — into an observable and scalable architecture. The platform is applied to the Pico dos Dias Observatory (OPD), contributing to the modernization of its infrastructure and enhancing capabilities for monitoring, remote diagnostics, and predictive maintenance. The project involves the development of Python-based components, integration of automated alerts, and virtualization using Docker, resulting in a functional, accessible, and reproducible solution applicable to other small- and medium-sized observatories.

Keywords: Telemetry, Messaging, Astronomical Observatories, InfluxDB, MQTT, Grafana, Modernization.

Sumário

1	Introdução	4
2	Objetivo	4
3	Materiais e Métodos	4
3.1	Revisão Bibliográfica e Definição da Arquitetura	4
3.2	Desenvolvimento de Simuladores de Sensores	5
3.3	Ambiente de Containerização e Testes de Conectividade	5
3.4	Definição da Arquitetura Final de Dados	5
3.5	Módulos de Comando e Supervisão	6
4	Resultados e Discussão	6
4.1	Implementação da Arquitetura de Dois Planos	6
4.2	Eficiência na Ingestão e Observabilidade de Dados	8
4.3	Integração de Alertas e Segurança Operacional	9
5	Conclusão	10

1 Introdução

Com o avanço das tecnologias de observação astronômica e a crescente complexidade dos instrumentos científicos, torna-se indispensável a modernização da infraestrutura de monitoramento e telemetria do Observatório do Pico dos Dias (OPD). A diversidade de subsistemas exige soluções integradas que possibilitem aquisição, armazenamento e visualização eficiente e acessível dos dados operacionais em tempo real.

Nesse contexto, este trabalho compõe o desenvolvimento de uma infraestrutura modular de telemetria, baseada em protocolos modernos de mensageria e bancos de dados de séries temporais, com foco na automação, escalabilidade e confiabilidade do sistema. Buscando reduzir o tempo despendido em tarefas operacionais, diagnósticos manuais e ajustes técnicos, permitindo que pesquisadores e equipes técnicas concentrem seus esforços na análise científica e na tomada de decisões estratégicas.

Assim, o projeto visa não apenas modernizar o processo de monitoramento do OPD, mas também tornar os dados mais acessíveis, organizados e visualmente interpretáveis, contribuindo para uma operação mais eficiente e alinhada às boas práticas internacionais em observatórios astronômicos.

2 Objetivo

O objetivo deste trabalho foi desenvolver uma infraestrutura de telemetria voltada ao monitoramento em tempo real dos subsistemas OPD, utilizando tecnologias modernas e de código aberto, de forma a tornar o sistema mais integrado, confiável e escalável.

Foi projetada uma arquitetura modular para o sistema de telemetria, definindo componentes independentes para aquisição, roteamento, armazenamento e visualização de dados. Também foi a implementação de rotinas em Python para coleta e envio de dados por meio de protocolos como MQTT, bem como a configuração do banco de dados InfluxDB para armazenamento eficiente de séries temporais.

Além disso, foram desenvolvidos dashboards interativos no Grafana para visualização em tempo real das informações monitoradas, implementados mecanismos de alertas automáticos para identificação de eventos críticos, realizados testes de integração e validação do sistema, e elaborada a documentação técnica dos componentes desenvolvidos, garantindo sua manutenção e possível expansão futura.

3 Materiais e Métodos

3.1 Revisão Bibliográfica e Definição da Arquitetura

Inicialmente foi realizada uma revisão bibliográfica sistemática de artigos técnicos sobre arquiteturas de telemetria aplicadas em observatórios astronômicos de médio e grande porte. Esta etapa permitiu identificar padrões de projeto, fundamentando a definição da arquitetura inicial do fluxo de dados.

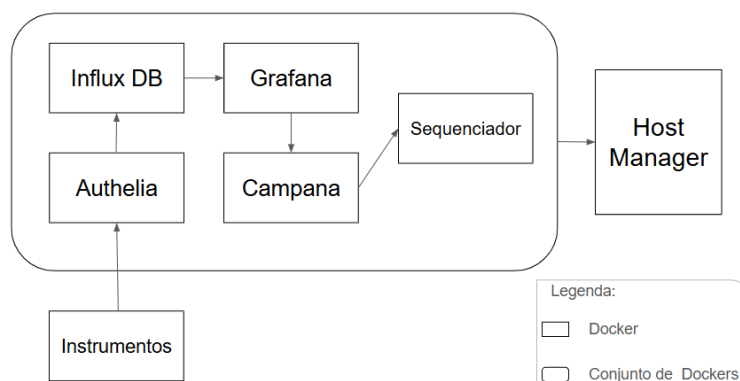


Figura 1: Arquitetura Inicial

3.2 Desenvolvimento de Simuladores de Sensores

Pela complexidade do ambiente do OPD, identificou-se a necessidade estratégica de desenvolver simuladores para validar os componentes do fluxo de dados sem comprometer a operação dos instrumentos reais.

1. Simulador de Validação Inicial: Desenvolvido em Python, utilizou tipos de dados simplificados para validar a conectividade e assegurar a familiarização com as ferramentas InfluxDB e Grafana.
2. Simuladores de Protocolo: Implementaram-se rotinas avançadas integrando protocolos de mensageria assíncrona (ZeroMQ e MQTT), permitindo o teste de novas configurações.
3. Simulador de Padrão Astronômico (ASCOM Alpaca): Para garantir a funcionalidade, foi desenvolvida uma simulação baseada no padrão industrial ASCOM Alpaca. Através da AlpacaAPI, o sistema passou a processar metadados idênticos aos gerados pelo hardware do observatório, garantindo a fidelidade dos testes de integração. funcionalidade?

3.3 Ambiente de Containerização e Testes de Conectividade

A infraestrutura foi isolada utilizando a tecnologia Docker. Inicialmente, instanciou-se um container primário contendo apenas o InfluxDB e o Grafana para validar a persistência dos dados e a configuração de dashboards preliminares.

Após esse desenvolvimento primário foi realizado um container possuindo o protocolo de mensageria e já implementado o Loki e o sequenciador. Fazendo assim o teste de conectividade desse tipo de dados e a visualização dos logs completos pelo Loki.

Por fim, foi desenvolvido um container com toda a arquitetura, afim de assim verificar a funcionalidade e identificar possíveis erros. E assim finalmente conectar ao Host Manager. funcionalidade?

3.4 Definição da Arquitetura Final de Dados

Durante a fase de integração, realizaram-se ajustes finos na arquitetura baseados em gargalos de conectividade identificados. A ferramenta de autenticação original (Authelia) mostrou-se excessivamente robusta para a camada de ingestão, sendo retirada do projeto até o atual momento. Para a segurança, implementou-se um Proxy Reverso via Nginx, garantindo proteção com menor impacto na latência.

A estratégia de observabilidade foi bifurcada para otimizar o desempenho:

- InfluxDB: Destinado exclusivamente ao armazenamento e visualização de séries temporais numéricas (métricas).
- Loki: Introduzido como solução para o armazenamento e análise de logs completos do sistema, permitindo a rastreabilidade de mensagens diretas e eventos de depuração.
- Telegraf: Implementado para padronização da coleta de logs e métricas de sistema.

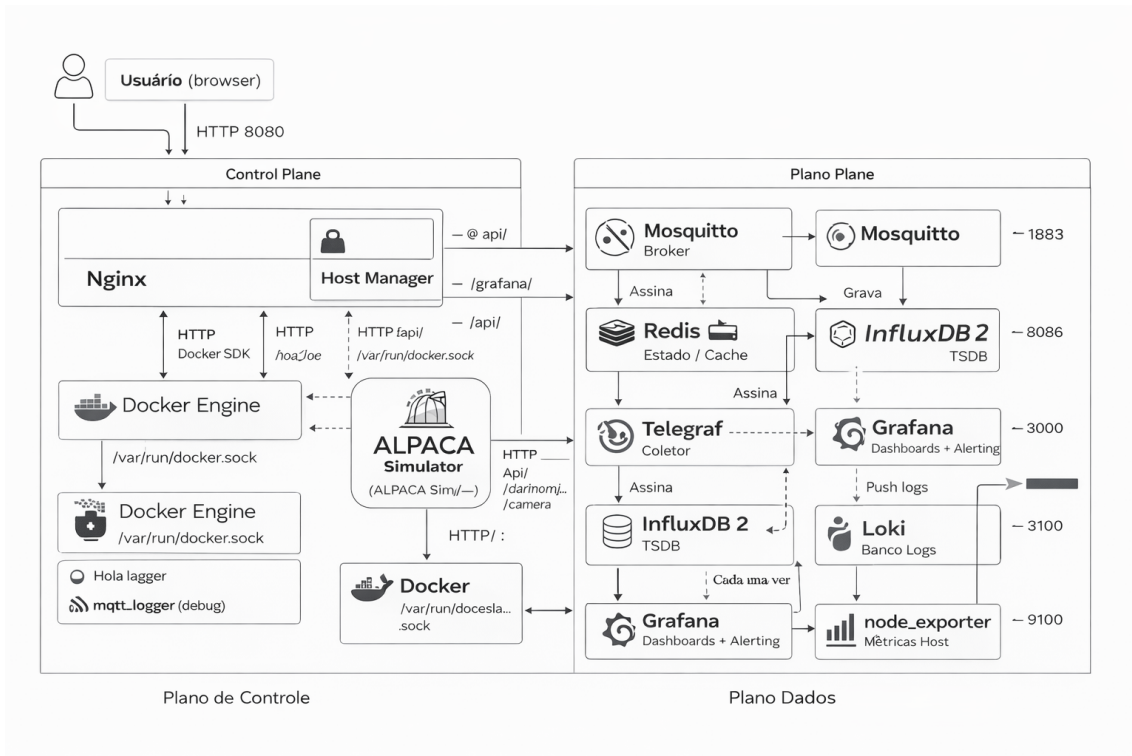


Figura 2: Arquitetura final de dados

3.5 Módulos de Comando e Supervisão

A lógica central da infraestrutura foi consolidada no Sequenciador, módulo responsável pela orquestração do fluxo de dados e comunicação com o Host Manager. O Host Manager atua como a interface de supervisão final, permitindo ao operador o acesso simultâneo aos dashboards do Grafana, ao monitoramento da saúde do sistema e à execução de comandos remotos nos instrumentos astronômicos.

4 Resultados e Discussão

4.1 Implementação da Arquitetura de Dois Planos

O principal resultado alcançado foi a estruturação de uma arquitetura funcional dividida em dois fluxos principais: o Plano de Controle e o Plano de Dados. A implementação bem-sucedida do gateway Nginx permitiu o roteamento centralizado para os diferentes serviços (Host Manager, Grafana e Sequencer), garantindo segurança via BasicAuth e simplificando o acesso externo através de uma porta única (8080).

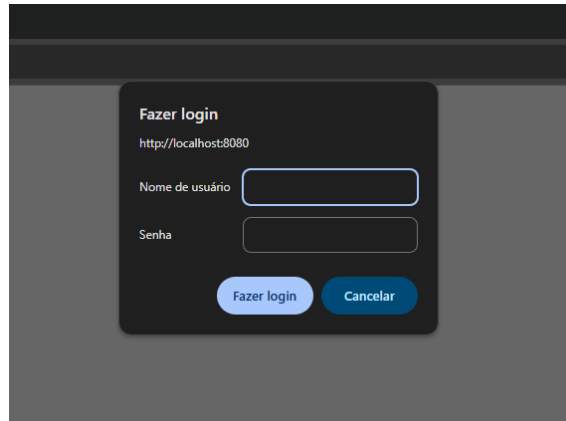


Figura 3: Autenticação

No Plano de Controle, observou-se que a integração entre o Host Manager e o Sequencer reduziu significativamente o tempo de resposta operacional para comandos críticos, como a abertura e o fechamento do domo e o início de exposições da câmera. A utilização do Redis como camada de cache e gerenciamento de estados provou ser fundamental para manter a persistência de variáveis operacionais e garantir a rapidez das requisições, evitando a perda de contexto entre processos do Sequencer.

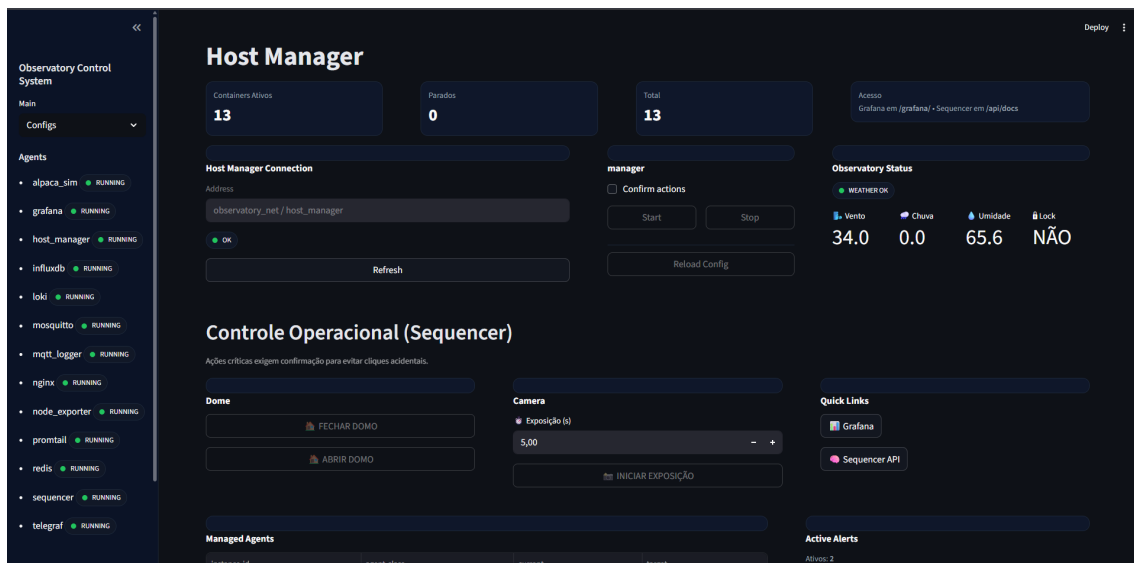


Figura 4: Host Manager 1

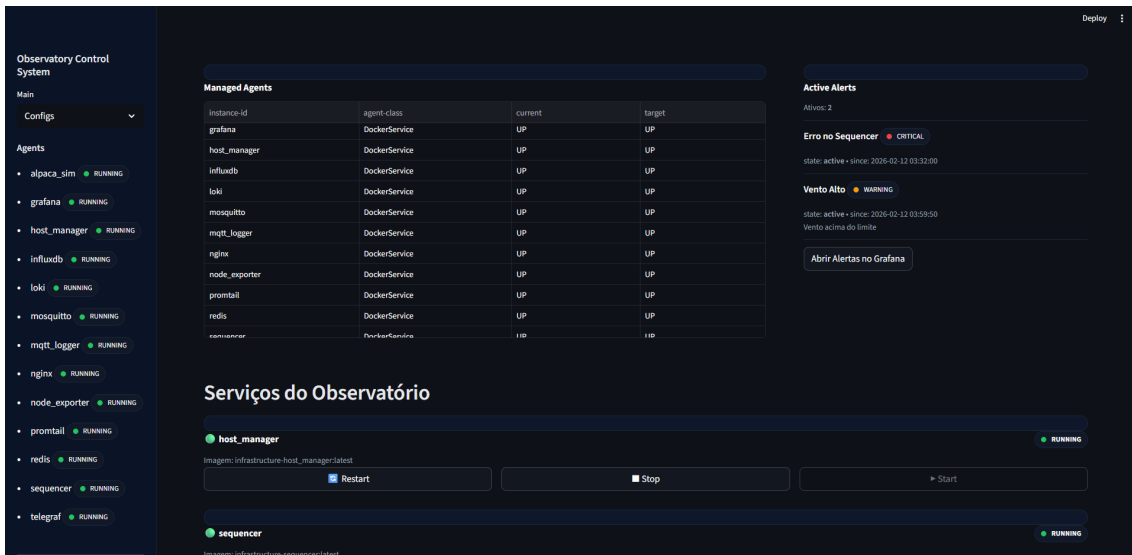


Figura 5: Host Manager 2

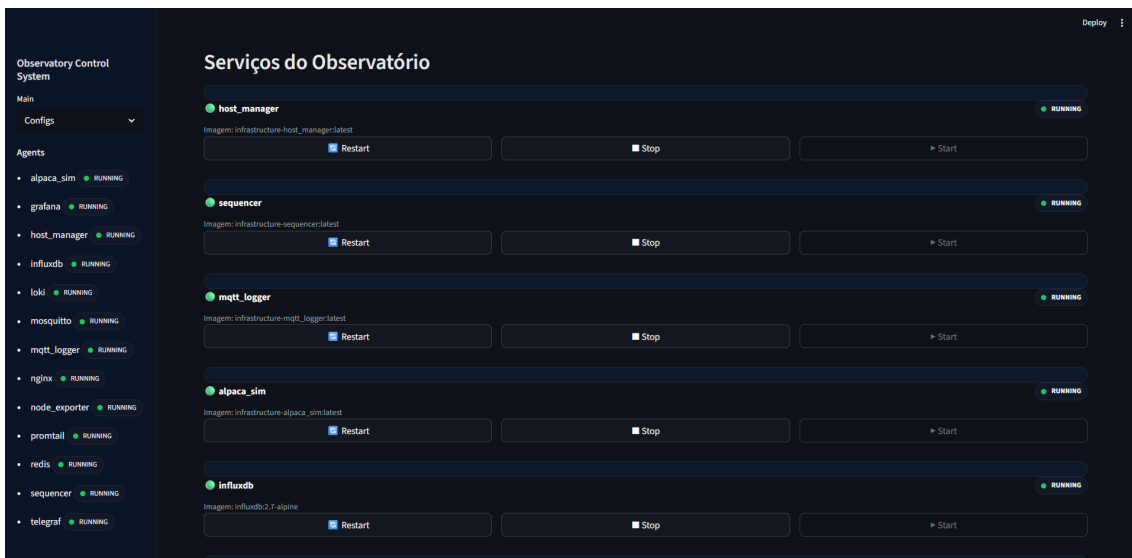


Figura 6: Host Manager 3

4.2 Eficiência na Ingestão e Observabilidade de Dados

No Plano de Dados, a cadeia de telemetria baseada em MQTT e Telegraf demonstrou alta confiabilidade no processamento de eventos. A discussão técnica revelou que o InfluxDB atende perfeitamente à demanda de séries temporais numéricas, permitindo a criação de gráficos de tendências para variáveis ambientais. Entretanto, a introdução do Grafana Loki sanou uma limitação crítica identificada nos testes iniciais: a incapacidade do InfluxDB de gerenciar de forma eficiente mensagens de log extensas e dados não estruturados. Através do Promtail, foi possível centralizar o troubleshooting de todos os containers Docker em uma única interface, elevando o grau de maturidade da manutenção do sistema.



Figura 7: Visualização dos dados do InfluxDB em Dashboard no Grafana

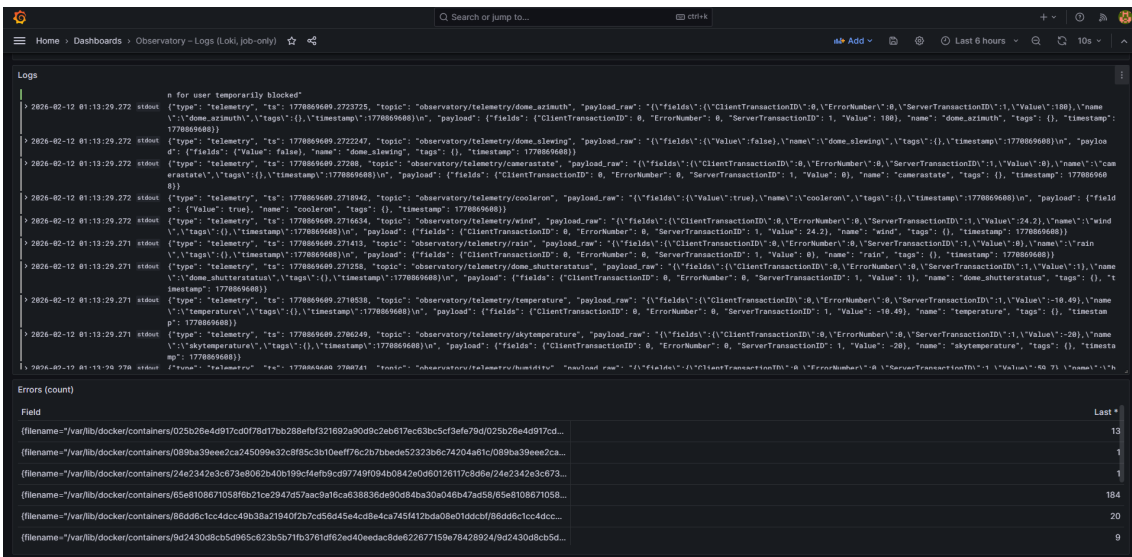


Figura 8: Visualização dos dados do Loki em Dashboard no Grafana

4.3 Integração de Alertas e Segurança Operacional

Um resultado de destaque foi a visualização de alertas ativos do Grafana diretamente na interface do Host Manager. Esta funcionalidade permitiu que o operador visualize estados de erro ou perigo sem a necessidade de alternar entre diferentes plataformas. A lógica de segurança implementada no Sequencer garante que comandos manuais sejam bloqueados automaticamente em caso de eventos climáticos adversos, utilizando os dados processados pelo InfluxDB como gatilho de decisão.

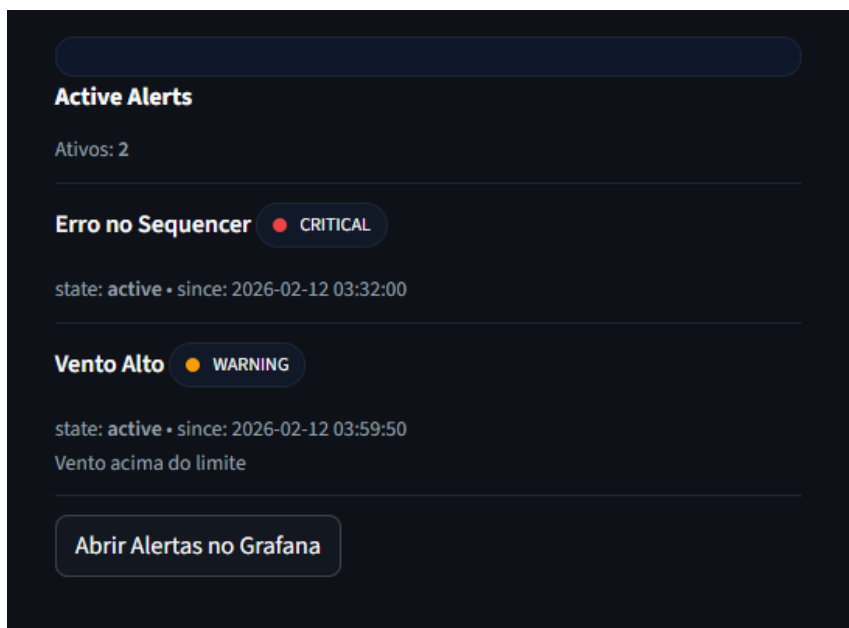


Figura 9: Alertas vindos do sistemas que são mandados pelo Grafana e recebidos pelo Host Manager

5 Conclusão

Os resultados obtidos demonstram que a arquitetura modular proposta cumpriu os requisitos de escalabilidade e confiabilidade necessários para o Observatório do Pico dos Dias. A substituição de processos manuais por um fluxo automatizado de telemetria e comando representa um avanço tecnológico significativo para a instituição. Conclui-se que a pesquisa teve seu objetivo cumprido, podendo evoluir para uma próxima fase de implementação no OPD.

Referências **Eu não encontrei no texto onde essas referencias são citadas**

GEND, C. van et al. The intelligent observatory: software for robotic operation, monitoring and safeguarding. In: SPIE. *Software and Cyberinfrastructure for Astronomy VIII*. [S.l.], 2024. v. 13101, p. 131012W.

HUERTAS, M. et al. Monitoring service for the gran telescopio canarias control system. In: SPIE. *Software and Cyberinfrastructure for Astronomy IV*. [S.l.], 2016. v. 9913, p. 99133P.

KOOPMAN, B. J. et al. The simons observatory: Deployment of the observatory control system and supporting infrastructure. In: SPIE. *Software and Cyberinfrastructure for Astronomy VIII*. [S.l.], 2024. v. 13101, p. 131014D.

NETO, A. F. et al. Sasquatch: Rubin observatory metrics and telemetry service. In: SPIE. *Software and Cyberinfrastructure for Astronomy VIII*. [S.l.], 2024. v. 13101, p. 131011M.

(KOOPMAN et al., 2024)

(NETO et al., 2024)

(HUERTAS et al., 2016)

(GEND et al., 2024)

**Falta escrever que o trabalho foi apresentado na SAB.
Falta uma seção de considerações finais e perspectivas.**