

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JOÃO MARCOS GUILHERME DE SOUZA

**VISUALIZAÇÃO E ANÁLISE DE DADOS PARA CIDADES INTELIGENTES: UM
ESTUDO COMPARATIVO ENTRE GRAFANA E KIBANA**

TOLEDO

2024

JOÃO MARCOS GUILHERME DE SOUZA

**VISUALIZAÇÃO E ANÁLISE DE DADOS PARA CIDADES INTELIGENTES: UM
ESTUDO COMPARATIVO ENTRE GRAFANA E KIBANA**

**DATA VISUALIZATION AND ANALYSIS FOR SMART CITIES: A
COMPARATIVE STUDY BETWEEN GRAFANA AND KIBANA**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Edson Tavares de Camargo

TOLEDO

2024



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

JOÃO MARCOS GUILHERME DE SOUZA

**VISUALIZAÇÃO E ANÁLISE DE DADOS PARA CIDADES INTELIGENTES: UM
ESTUDO COMPARATIVO ENTRE GRAFANA E KIBANA**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Tecnólogo em Tecnologia em Sistemas
para Internet do Curso Superior de Tecnologia
em Sistemas para Internet da Universidade
Tecnológica Federal do Paraná.

Data de aprovação: 06/setembro/2024

Ivan Luiz Salvadori
Doutor em Ciência da Computação
Universidade Tecnológica Federal do Paraná

Roberto Milton Scheffel
Doutor em Ciência da Computação
Universidade Tecnológica Federal do Paraná

Edson Tavares de Camargo
Doutor em Informática
Universidade Tecnológica Federal do Paraná

**TOLEDO
2024**

AGRADECIMENTOS

Agradeço a minha família e amigos que sempre estiveram comigo e me apoiaram. Em especial aos meus pais Renato e Rafaela, por me incentivar a sempre buscar conhecimento e sabedoria. E a minha irmã Gabriela por todo o carinho durante toda a minha jornada. A todos os meus professores, o conhecimento aqui expresso reflete o empenho e trabalho duro de vocês. Um agradecimento especial ao meu orientador Edson, por todos os ensinamentos compartilhados. Este trabalho não existiria sem vocês.

RESUMO

Com o crescente aumento no volume de dados gerados na sociedade contemporânea, a análise e visualização de dados são atividades cruciais para transformá-los em informações úteis. No contexto das cidades inteligentes, torna-se fundamental a escolha de ferramentas capazes de fornecer informações relevantes para os interessados e tomadores de decisão. Este estudo realiza uma análise comparativa das ferramentas Kibana e Grafana para visualização e análise de dados de aplicações de cidades inteligentes. Ambas as ferramentas, amplamente reconhecidas na comunidade de desenvolvimento de software, possuem recursos que permitem lidar com grandes volumes de dados e personalizar painéis de visualização. A pesquisa envolve a instalação, configuração e integração das ferramentas com as fontes de dados de aplicações de cidades inteligentes em desenvolvimento, a saber: monitoramento dos caminhões da coleta de lixo e monitoramento dos da qualidade do ar. Serão avaliados critérios como facilidade de instalação e integração e a personalização de painéis de visualização perante requisitos de software de cada aplicação. Este estudo contribui para o avanço da visualização de dados em cidades inteligentes, oferecendo informações que auxiliam na escolha e no uso adequado das ferramentas Kibana e Grafana, com potencial para melhorar as soluções tecnológicas de gerenciamento inteligente de cidades e, assim, a qualidade de vida dos habitantes. Os resultados apresentam que ambas as ferramentas têm capacidade de apresentar os dados de maneira clara para visualização, atendendo a maioria dos requisitos impostos. Contudo, do ponto de vista do usuário final, a parametrização e configuração não é intuitiva fazendo-se necessário auxílio de profissionais capacitados.

Palavras-chave: cidades inteligentes; análise de dados; visualização de dados; grafana; kibana.

ABSTRACT

With the growing volume of data generated in contemporary society, data analysis and visualization have become crucial activities for transforming data into useful information. In the context of smart cities, it is essential to choose tools capable of providing relevant insights to stakeholders and decision-makers. This study performs a comparative analysis of the Kibana and Grafana tools for data visualization and analysis in smart city applications. Both tools, widely recognized in the software development community, have features that handle large volumes of data and customize visualization dashboards. The research involves the installation, configuration, and integration of these tools with data sources from smart city applications under development, namely: monitoring of waste collection trucks and air quality monitoring. Criteria such as ease of installation and integration, and dashboard customization relative to the software requirements of each application will be evaluated. This study contributes to the advancement of data visualization in smart cities, providing information that aids in the choice and appropriate use of Kibana and Grafana tools, with the potential to enhance technological solutions for intelligent city management and thus improve the quality of life for residents. The results show that both tools can present data clearly for visualization, meeting most of the imposed requirements. However, from the end-user perspective, the configuration and parametrization are not intuitive, requiring the assistance of skilled professionals.

Keywords: smart cities; data analytics; data visualization; grafana; kibana.

LISTA DE TABELAS

Tabela 1 – Dados da aplicação 1	21
Tabela 2 – Dados da aplicação 2	22
Tabela 3 – Requisitos da Aplicação de Monitoramento da Coleta de Lixo	26
Tabela 4 – Requisitos da aplicação de monitoramento da qualidade do ar	26
Tabela 5 – Atendimento dos requisitos da aplicação 1	51
Tabela 6 – Atendimento dos Requisitos da aplicação 2	55

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	9
1.1.1	Objetivo geral	9
1.1.2	Objetivos específicos	10
1.1.3	Itens Fora do Escopo	10
1.2	Justificativa	10
1.3	Estrutura do trabalho	11
2	REFERENCIAL TEÓRICO	12
2.1	Definição de Cidades Inteligentes	12
2.1.1	Sentilo	12
2.1.2	Elastic Stack	12
2.2	Kibana	13
2.3	Grafana	14
2.4	Lucene	15
2.5	Trabalhos Relacionados	16
3	MATERIAIS E MÉTODOS	18
3.1	Materiais	18
3.1.1	Servidor	18
3.1.2	Programa Cidades Inteligentes em Toledo/PR	18
3.1.2.1	Aplicação 1: Monitoramento dos caminhões da coleta Seletiva	19
3.1.2.2	Aplicação 2: Monitoramento da Qualidade do Ar	19
3.2	Plataformas de Software para Cidades Inteligentes	20
3.2.1	Dados das aplicações	20
3.3	Métodos	22
3.3.1	Infraestrutura Básica	22
3.3.2	Instalação e Configuração do Grafana	23
3.3.3	Instalação e Configuração do Kibana	24
3.3.4	Requisitos da Aplicação 1: Monitoramento da Coleta de Lixo	25
3.3.5	Requisitos da aplicação 2: Monitoramento da Qualidade do Ar	26
4	RESULTADOS	28

4.1	Criação dos <i>dashboards</i> no Grafana	28
4.1.1	Painéis da aplicação 1 no Grafana	28
4.1.2	Painéis da aplicação 2 no Grafana	30
4.2	Criação dos <i>dashboards</i> no Kibana	35
4.2.1	Painéis da aplicação 1 no Kibana	35
4.2.2	Painéis da aplicação 2 no Kibana	37
4.3	Comparação com Requisitos Definidos para Aplicação 1	41
4.3.1	RF01: Armazenar as medições dos sensores das estações	41
4.3.2	RF02: Apresentar um mapa da cidade com todas as estações	42
4.3.3	RF03: Apresentar as medições dos sensores das estações	43
4.3.4	RF04: Apresentar o histórico de medições dos sensores das estações	44
4.3.5	RF05: Apresentar o Índice de qualidade do Ar (IQA) das estações	45
4.3.6	RF06: Apresentar o histórico do Índice de qualidade do Ar (IQA) das estações	46
4.3.7	RF07: Apresentar o Índice de calor das estações	47
4.3.8	RF08: Apresentar o histórico do Índice de calor das estações	47
4.3.9	RF09: Apresentar a localização do usuário no mapa	48
4.3.10	RF10: Apresentar os melhores e piores índices e parâmetros das últimas 24 horas	48
4.3.11	RF11: Apresentar o histórico dos melhores e piores índices e parâmetros	50
4.3.12	RF12: Apresentar o dia da semana com os melhores e piores índices e parâmetros	50
4.3.13	RF13: Apresentar o horário com os melhores e piores índices e parâmetros para cada estação	50
4.4	Comparação com Requisitos Definidos para Aplicação 2	51
4.4.1	RF01: Apresentar um mapa da cidade com todos os veículos	51
4.4.2	RF02: Apresentar a localização do usuário no mapa	53
4.4.3	RF03: Apresentar a localização de veículo específico no mapa	53
4.4.4	RF04: Apresentar o histórico de rotas dos veículos	53
4.4.5	RF05: Apresentar o histórico de coleta/passagem de veículo em determinada localização	54
4.5	Avaliação das Documentações das Ferramentas	56
4.6	Externalização de Dashboards	56

5	CONCLUSÃO	59
5.1	Trabalhos Futuros	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

Com o avanço contínuo da tecnologia e a crescente urbanização, as cidades enfrentam desafios complexos e demandas cada vez maiores para se adaptar a um ambiente em constante mudança. Nesse contexto, o conceito de Cidades Inteligentes surgiu como uma resposta à necessidade de melhorar a qualidade de vida dos cidadãos, otimizar recursos e promover o desenvolvimento sustentável. Uma cidade inteligente é um ambiente urbano que utiliza amplamente as Tecnologias da Informação e Comunicação (TIC) para otimizar o gerenciamento de ativos e recursos em diferentes setores, como transporte, saúde, educação, tecnologia da informação e controle ambiental. (KIRIMTAT *et al.*, 2020; SÁNCHEZ-CORCUERA *et al.*, 2019; SANTANA *et al.*, 2017).

Uma questão central ao fazer uso de aplicações voltadas às cidades inteligentes reside na análise e apresentação dos dados. Os dados coletados por diversos sensores espalhados na cidade ou mesmo coletados dos *smartphones* dos habitantes são transmitidos para a Nuvem e armazenados em grandes bases de dados. Ocorre que os dados coletados ganham valor a medida que trazem informações relevantes tanto para gestores quanto para os cidadãos. Esses dados, se bem utilizados, podem vir a alavancar políticas públicas para melhorar a qualidade de vida (KIRIMTAT *et al.*, 2020). Dessa forma, o problema central que este trabalho pretende abordar reside em pesquisar ferramentas abertas e gratuitas para auxiliar na exploração e visualização de grandes volumes de dados.

A literatura especializada (PRIAMBODO; KADARINA, 2020; KHANH, 2021; BARRO *et al.*, 2019) apresenta as ferramentas Grafana¹ e Kibana² como duas importantes ferramentas abertas e gratuitas utilizadas para visualizar dados para soluções em cidades inteligentes. Priambodo e Kadarina (2020) usaram a ferramenta Kibana em um sistema de monitoramento para pacientes com COVID-19 em auto-isolamento. Kibana ajudou na análise e visualização de dados de dispositivos médicos conectados, permitindo o monitoramento contínuo da condição dos pacientes e a implementação oportuna de medidas preventivas para promover sua recuperação. Khanh (2021), da Universidade Tecnológica de Yangon, Myanmar, utilizaram a ferramenta Grafana para monitorar a qualidade do ar e a temperatura local. Grafana foi responsável pela consulta e visualização dos dados por meio de gráficos.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral deste trabalho é comparar as ferramentas Kibana e Grafana a fim de identificar as vantagens e desvantagens de sua utilização para uso em aplicações de software

¹ <https://grafana.com/grafana>

² <https://www.elastic.co/pt/kibana/>

em cidades inteligentes. Os parâmetros de comparação incluem a qualidade da documentação, facilidade de instalação e a necessidade de atender os requisitos das aplicações de monitoramento de qualidade do ar e monitoramento dos caminhões da coleta seletiva, em desenvolvimento no contexto de uma parceria entre o Município de Toledo e a Universidade. (ROSSATO; SPANHOL; CAMARGO, 2020) (PASTÓRIO *et al.*, 2022).

1.1.2 Objetivos específicos

Para atingir o objetivo proposto, os seguintes objetivos específicos devem ser alcançados:

- Avaliar a qualidade da documentação das ferramentas.
- Analisar a facilidade de instalação e configuração para o uso das ferramentas.
- Comparar a capacidade de personalização dos painéis de visualização de cada ferramenta diante dos requisitos de software das aplicações mencionadas.

1.1.3 Itens Fora do Escopo

Deve ser destacado que para alcançar o objetivo proposto não será incluído no escopo deste trabalho:

- Desenvolvimento ou implementação de novas aplicações de cidades inteligentes.
- Configuração da plataforma de software responsável por integrar sensores e bases de dados. Neste trabalho assume-se que a plataforma Sentilo e a base de dados Elasticsearch já estão configuradas e em uso. De fato, a equipe do projeto mantém a plataforma e base de dados operacionais.
- Requisitos das consultas necessárias para cada aplicação. Os requisitos serão adotados de um trabalho de mestrado em andamento no contexto das pesquisas em cidades inteligentes.
- Configurações avançadas de segurança ou privacidade dos dados.

1.2 Justificativa

A escolha apropriada de uma ferramenta de visualização de dados desempenha um papel importante na compreensão das informações coletadas e na tomada de decisões. Essa análise permitirá identificar as capacidades, características e possíveis vantagens de cada ferramenta.

As aplicações selecionadas para avaliação fazem parte de um projeto de colaboração entre a UTFPR Campus Toledo e a Prefeitura do Município de Toledo para desenvolver um projeto de Cidades Inteligentes (Universidade Tecnológica Federal do Paraná, 2020). A primeira delas aborda o rastreamento online dos caminhões de coleta seletiva (ROSSATO; SPANHOL; CAMARGO, 2020), fornecendo informações para otimizar a eficiência deste serviço. A segunda aplicação está voltada para o monitoramento da qualidade do ar (PASTÓRIO *et al.*, 2022), permitindo um acompanhamento da poluição atmosférica e seus impactos na saúde pública e meio ambiente. Ambas as aplicações serão fontes de dados que serão explorados através das ferramentas Grafana e Kibana, dados estes que foram e são coletados nos trabalhos citados. Também utilizou-se requisitos pré-estabelecidos que definem o que cada aplicação deve apresentar para ser realmente útil e relevante no contexto de cidades inteligentes. Os requisitos serviram como métrica para avaliar e comparar a eficácia na visualização e análise dessas informações nas ferramentas Grafana e Kibana, contribuindo assim para o avanço das soluções tecnológicas em cidades inteligentes.

1.3 Estrutura do trabalho

O restante deste trabalho está estruturado da seguinte forma: o Capítulo 2 traz o referencial teórico apresentando as tecnologias adotadas bem como informações de contextualização deste trabalho. Já no Capítulo 3 são apresentadas as tecnologias utilizadas e a metodologia empregada para tanto. No capítulo 4 são abordados os resultados obtidos e por fim o capítulo 5 expõe as considerações finais.

2 REFERENCIAL TEÓRICO

Este Capítulo aborda as tecnologias relacionadas com este trabalho. Inicialmente, estão descritos os conceitos de cidades inteligente bem como as aplicações que serão utilizadas como fonte de dados para a pesquisa. Logo após, estão apresentados as tecnologias que formam a infraestrutura das aplicações, cujo captam e armazenam os dados. E por fim uma descrição das ferramentas que são base deste trabalho: Grafana e Kibana.

2.1 Definição de Cidades Inteligentes

Como o objetivo principal é a comparação das ferramentas de visualização Grafana e Kibana, foi necessário a obtenção de dados de alguma base de dados já estabelecida e que possui uma considerável quantidade de dados. Por tanto, duas aplicações foram escolhidas: uma aplicação de rastreamento de veículos (CAMARGO; SPANHOL; SOUZA, 2021) e outra aplicação de monitoramento de qualidade do ar (PASTÓRIO *et al.*, 2022).

2.1.1 Sentilo

O principal componente responsável pela coleta de dados do programa Cidades Inteligentes no município de Toledo é a plataforma Sentilo. A Sentilo é um elemento crucial na arquitetura de Cidade Inteligentes, isolando os aplicativos desenvolvidos para explorar as informações geradas pela cidade, juntamente com a camada de sensores distribuídos por toda a cidade para coleta e disseminação dessas informações. (SINAEEPOURFARD *et al.*, 2016)

Neste trabalho, a plataforma Sentilo é responsável coleta dados de diversos sensores espalhados pela cidade de Toledo, registrando milhares de *logs* diários em sua base de dados. A plataforma é organizada em três camadas principais: Aquisição de Dados, Transmissão de Dados e Armazenamento dos Dados. A primeira camada coleta dados brutos de provedores de dados, que no caso das aplicações selecionadas são sensores instalados nas estações de monitoramento da qualidade do ar, e os sensores instalados nos caminhões de lixo. A segunda camada é responsável pela transferência desses dados brutos da camada de aquisição para o Middleware, utilizando métodos de comunicação como LoraWan. A terceira camada é responsável por armazenar os dados obtidos no banco de dados Elasticsearch.

2.1.2 Elastic Stack

Elastic Stack é um nome comumente utilizado para os produtos desenvolvidos pela Elastic - Elasticsearch, Logstash, Kibana¹. É um conjunto de poderosos e complexos softwares ana-

¹ <https://www.elastic.co/what-is/elk-stack>

líticos. Cada uma dessas aplicações é de código aberto e está disponível no repositório do GitHub. Empresas como Netflix, Adobe, Walmart e Audi (ELASTIC, 2021a) usam essas ferramentas para processar bilhões de pontos de dados e criar painéis analíticos.

A parte principal deste sistema é o Elasticsearch, um mecanismo de pesquisa que utiliza um banco de dados orientado a documentos no formato JSON. Sua linguagem de desenvolvimento principal é o Java, e o aspecto central é o rápido acesso aos dados predefinidos. É um dos pedaços de software de pesquisa mais eficientes com a possibilidade de consulta de texto completo. O Elasticsearch fornece todas essas funcionalidades por meio de uma API REST, embora muitas linguagens de programação cubram essa conexão com suas bibliotecas (ELASTIC, 2021b).

O Logstash é outra parte essencial do Elastic Stack. Ele transfere e processa dados brutos para o Elasticsearch. O pré-processamento é definido pelo arquivo de configuração do Logstash, que contém as especificações de filtro e transformação. Essa ferramenta pode transportar dados gravados no passado para o Elasticsearch, bem como logs de sistemas. O principal benefício do desenvolvimento de configurações complexas do Logstash é uma definição de consulta de pesquisa significativamente melhor e menos modificação de dados na conexão do Kibana (ELASTIC, 2021d).

A última ferramenta no acrônimo é o Kibana, que visualiza os pontos de dados do Elasticsearch. O desenvolvimento do Kibana é realizado agora no framework React, em vez do AngularJS, que era usado no passado. A arquitetura de componentes permite expandir a aplicação com plugins de servidor ou frontend. Muitos projetos de plugins de visualização estão disponíveis no GitHub. Em combinação com gráficos integrados, ele cria painéis valiosos que mostram características essenciais dos dados. O guia de desenvolvimento bem escrito do Kibana é benéfico para qualquer trabalho adicional no núcleo da aplicação e plugins personalizados (ELASTIC, 2021c).

Neste trabalho o Elasticsearch é utilizado pois é a base dados utilizada pelas aplicações de cidades inteligentes. E a Kibana por se tratar de uma das ferramentas motivadoras para este trabalho.

2.2 Kibana

Como citado anteriormente o Kibana é uma ferramenta que faz parte da Elastic Stack. O Kibana foi projetado como uma plataforma de visualização para uso exclusivo em conjunto com o Elasticsearch (ELASTIC, 2021c). Ele fornece uma interface baseada na web para pesquisar, visualizar e analisar os dados armazenados em um cluster do Elasticsearch. A visualização principal do Kibana é dividida em quatro componentes principais: Descoberta (Discover), Visualização (Visualize), Painéis (Dashboards) e Gerenciamento (Management) (BAJER, 2017).

A seção de Gerenciamento é onde todas as configurações internas do Kibana podem ser configuradas. É o local onde os padrões de índice devem ser definidos. Se os padrões de

índice contiverem eventos baseados em horário, também é necessário especificar o campo de data e hora pelo qual o Kibana classificará e filtrará os dados. Com base em um conjunto de índices que atendem ao padrão de índice selecionado, o Kibana mostra a lista de campos do índice juntamente com seu tipo e propriedades. Além disso, é possível modificar formatações de campo para alterar como o valor do campo é exibido na interface do Kibana.

A função Descoberta permite navegar e analisar interativamente as entradas de dados puras. Os documentos do Elasticsearch são categorizados com base nos padrões de índice definidos na seção de Gerenciamento. Eles podem ser facilmente pesquisados e filtrados por horário ou por propriedades do documento usando a sintaxe de consulta do Apache Lucene. Também é possível ver o número de documentos que correspondem à consulta de pesquisa ou obter estatísticas de valores de campos.

Devido à arquitetura baseada em plugins, o Kibana pode ser facilmente estendido para atender a necessidades específicas. A seção Visualização fornece a possibilidade de visualizar dados com um dos plugins de visualização fornecidos. As informações podem ser exibidas na forma de tabelas, gráficos, mapas, histogramas e muitos outros formatos. Grandes volumes de dados podem ser facilmente mostrados na forma de gráficos de pizza, gráficos de barras, gráficos de linha ou gráficos de dispersão.

O Painel permite combinar vários resultados salvos da Descoberta e Visualização em uma única visualização. É possível organizar e redimensionar os elementos do painel conforme necessário. Com alguma experiência, é possível criar painéis muito sofisticados e coloridos diretamente na interface do Kibana. Os painéis podem ser facilmente salvos, compartilhados ou incorporados em outras páginas da web.

2.3 Grafana

Grafana é uma plataforma de código aberto que fornece recursos para consulta, visualização e entendimento de métricas (GRAFANA, 2023). Grafana ajudou a responder as perguntas de como visualizar dados de maneira simples e prática, o que tem sido um desafio para muitos desenvolvedores nos últimos anos. Essa ferramenta pode reunir todos os dados de diferentes fontes e os compartilha entre todas as partes interessadas. Ela também conta com vários *plugins*, cujo os usuários podem obter o que mais lhes convém ou até mesmo criar *plugins* para atender às suas necessidades (KHANH, 2021). Por ser de código aberto, o Grafana é apoiado por uma comunidade muito ativa, com empenho em torna-lo melhor.

Grafana oferece os seguintes recursos (Grafana Document, 2021) :

- Visualização de Dados: Gráficos rápidos e flexíveis do lado do cliente com muitas opções. A criação dos painéis possibilita o desenvolvedor a deixar os dados da maneira que ele quiser, seja mostrar apenas um campo da consulta ou o *log* completo.

- *Dashboard Dinâmico*: *Dashboards* dinâmicos reutilizáveis com vários painéis. Além disso, é possível criar variáveis personalizadas no seu *dashboard*, e inseri-las diretamente no campo de busca.
- Explorar Métricas: Comparar diferentes fontes de dados e consultas dividindo visualizações. É possível utilizar várias fontes de dados em um mesmo *dashboard*, basta configurar as fontes corretamente.
- Alertas: É possível configurar para que determinadas métricas gerem alertas quando atingem um certo valor, as alertas podem ser recebidas via *e-mail* e diretamente no *dashboard*.
- Filtros Ad-hoc: Criando novas chaves-valor na hora, que serão aplicados automaticamente a todas as consultas que usam essa fonte de dados usando filtros Ad-hoc.

Além de fornecer um grande número de visualizações e fontes de dados apenas com a versão padrão do Grafana, podemos usar os *plugins* que estendem a funcionalidade dele. Podemos instalar *plugins* construídos pela comunidade ou criar um. O Grafana suporta três tipos de *plugins*: painéis, fontes de dados e aplicativos.

- *Plugins* de fonte de dados: fontes de dados externas não suportadas inicialmente, podem interagir com um *plugin* personalizado e transformar os dados em um formato compreensível pelo Grafana (CHAKRABORTY; KUNDAN, 2021). É possível começar a usar os dados facilmente em qualquer painel de visualização, adicionando uma fonte de dados com o *plugin* criado.
- Aplicativos: Fontes de dados e painéis são combinados usando o *plugin* de aplicativos para fornecer uma experiência de usuário consistente (Grafana Document, 2021). *Plugins* de aplicativos são usados para criar uma experiência de monitoramento personalizada.
- Painéis: O *dashboard* pode ter novos tipos de visualização adicionando um *plugin* de painel. Neste tipo de *plugin* é possível criar novos designs de painéis com cores e formatos diferentes dos padrões. É utilizado para visualizar dados retornados por consultas de fonte de dados, navegar entre *dashboards* ou controlar sistemas externos (CHAKRABORTY; KUNDAN, 2021).

2.4 Lucene

O Apache Lucene é uma biblioteca em Java projetada para realizar indexação e busca de informações com base na similaridade de textos. Entre os principais benefícios de seu uso, destacam-se a escalabilidade na recuperação de dados, alto desempenho, simplicidade de uso

e o fato de ser uma ferramenta de código aberto. Entretanto, para seu funcionamento, é essencial que haja informações derivadas de conteúdo textual, independentemente da linguagem ou formato do arquivo em que estejam armazenadas (LAKHARA; MISHRA, 2017).

A Lucene disponibiliza uma API que permite ao programador indexar e buscar textos de maneira simples, sem a necessidade de compreender toda a complexidade técnica envolvida. Em outras palavras, a complexidade do processo de indexação é abstraída, permitindo que o desenvolvedor foque nas regras de negócio da aplicação.

Neste trabalho, a Lucene foi utilizada de maneira interna nas ferramentas Grafana e Kibana para busca e filtragem dos dados para obter as consultas desejadas.

2.5 Trabalhos Relacionados

Este capítulo tem como objetivo apresentar publicações cujas temáticas se correlacionam à deste estudo. Além disso, ressaltar os resultados e os progressos das pesquisas estudadas e relacioná-los com a proposta de apresentação deste trabalho.

Thu *et al.* (2018), da universidade tecnológica de Yangon, Myanmar, fez uso do Grafana para monitorar a qualidade do ar e a temperatura local. Eles captavam os dados a partir de uma rede LoraWAN e inseriram em uma base de dados InfluxDB, então o Grafana fazia as consultas e era responsável pela disponibilidade de visualização dos dados por meio de gráficos.

Priambodo e Kadarina (2020), utilizaram o Kibana como parte integrante de um sistema de monitoramento para pacientes com COVID-19 em auto-isolamento. A ferramenta desempenhou um papel importante na análise e visualização dos dados coletados pelos dispositivos médicos conectados à Internet, que registravam medidas fisiológicas, como saturação de oxigênio no sangue (SpO2) e frequência cardíaca, juntamente com informações de localização dos pacientes. Essa abordagem permitiu o monitoramento contínuo da condição dos pacientes, possibilitando a adoção de ações preventivas de forma oportuna, visando sua recuperação.

O trabalho de Chan (2019) utilizou o Grafana para coletar, integrar e visualizar estatísticas de utilização de clusters de computação de alto desempenho. Diversas fontes de dados foram consolidadas, incluindo informações de jobs do Slurm, associação de contas e métricas de CPU coletadas com Telegraf. Esses dados foram integrados e apresentados de forma visual, permitindo uma análise mais fácil e identificação de tendências. O objetivo era facilitar a tomada de decisões e a alocação eficiente de recursos através da visualização clara das estatísticas de utilização do cluster.

No projeto de monitoramento de estufa controlada de Sarturi (2022), foi utilizado o Grafana como ferramenta para o tratamento e visualização dos dados. Os dados ambientais, como temperatura, umidade, luminosidade, CO2 e qualidade do ar, foram armazenados em um banco de dados MySQL e visualizados online por meio de um dashboard. Esse dashboard permitiu que os usuários acompanhassem de forma intuitiva as condições dentro da estufa e tomassem medidas de controle, se necessário, utilizando os atuadores disponíveis. Com isso, o Grafana

contribuiu significativamente para o monitoramento eficaz do ambiente da estufa, fornecendo uma ferramenta poderosa para a gestão e otimização do cultivo protegido.

O Kibana foi empregado como a ferramenta de visualização dos dados coletados pela arquitetura de monitoramento de eventos de segurança cibernética para sistemas industriais desenvolvido por Barboza (2020). A arquitetura desenvolvida coletou dados de diversos objetos monitorados, como arquivos de log, tráfego de rede e tentativas de intrusão, e os armazenou em um banco de dados Elasticsearch. O Kibana foi utilizado para criar visualizações gráficas e dashboards interativas com base nesses dados. Essas visualizações forneceram uma representação visual dos eventos de segurança cibernética, facilitando a detecção de anomalias e ciberataques pelos profissionais da área.

3 MATERIAIS E MÉTODOS

Neste Capítulo, são apresentados os recursos e a abordagem que utilizada para atingir os objetivos deste estudo. A Seção 3.1 dedica-se a apresentar as principais tecnologias empregadas para o desenvolvimento deste trabalho e a Seção 3.3 demonstra como as tecnologias foram empregadas para realização do trabalho.

3.1 Materiais

3.1.1 Servidor

Para o desenvolvimento deste trabalho foi utilizado um servidor UTFPR Campus Toledo disponibilizado pelo grupo de pesquisa relacionado ao projeto. O servidor já estava configurado e preparado para receber dados de aplicações em operação de cidades inteligentes através da plataforma Sentilo (PAGANINI, 2024). Como ferramenta de containerização optou-se por empregar o Docker¹, uma plataforma amplamente reconhecida, para facilitar o processo de instalação e execução dessas ferramentas. O Docker oferece uma solução versátil e eficiente para a implantação de aplicativos em ambientes isolados, conhecidos como contêineres, tornando mais fácil a gestão de dependências e a garantia de consistência nas configurações.

Grafana e Kibana foram instaladas e configuradas através do Docker disponível no DockerHub², um repositório central de imagens de contêineres. Essas imagens pré-configuradas oferecem uma maneira conveniente de configurar as ferramentas de análise de dados, economizando tempo e recursos. Além disso, a facilidade de distribuição e replicação das configurações do ambiente de desenvolvimento tornou o Docker uma escolha lógica para este projeto.

A Figura 1 apresenta a disposição dos contêineres das aplicações em execução no servidor, onde é possível observar as ferramentas Grafana e Kibana com comunicação direta a base de dados Elasticsearch.

3.1.2 Programa Cidades Inteligentes em Toledo/PR

As aplicações escolhidas fazem parte de um projeto de colaboração em cidades inteligentes entre a UTFPR Campus Toledo e a Prefeitura do Município de Toledo (Universidade Tecnológica Federal do Paraná, 2020), que tem como objetivo melhorar a qualidade de vida dos habitantes através da implantação de uma rede sem-fio de longo alcance e baixa potência (ROSSATO; SPANHOL; CAMARGO, 2020).

¹ <https://docs.docker.com/>

² <https://hub.docker.com/>

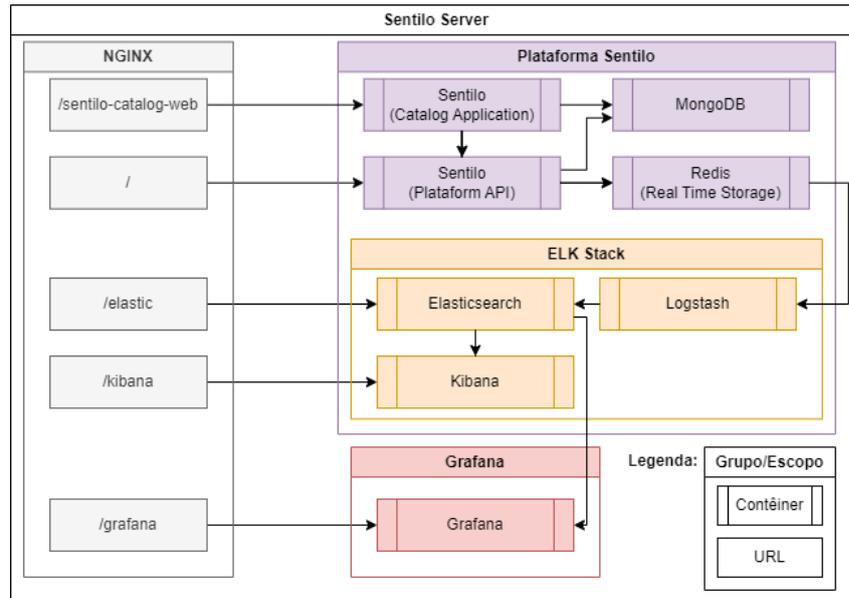


Figura 1 – Servidor Sentilo

Fonte: Conteúdo adaptado de Paganini (2024)

3.1.2.1 Aplicação 1: Monitoramento dos caminhões da coleta Seletiva

A aplicação de rastreamento dos caminhões de coleta seletiva tem como objetivo principal fornecer informações *online* sobre a posição dos caminhões, permitindo o monitoramento e controle eficiente do serviço (CAMARGO; SPANHOL; SOUZA, 2021).

Através de dispositivos de hardware, como o TTGO TBean 1.10, juntamente com um Arduino, GPS e rádio LoRa, os caminhões de coleta seletiva enviam dados de localização, como latitude, longitude, precisão do GPS (HDOP), velocidade e altitude. Esses dados são enviados para um servidor usando a tecnologia LoRaWAN, utilizando a infraestrutura da *The Things Network* (TTN). E por fim são inseridos em uma base de dados Elasticsearch, que é de onde foram consumidos pelas ferramentas Grafana e Kibana para o desenvolvimento deste trabalho.

Com base nos dados gerados, é possível realizar diversas consultas relevantes. Essas consultas incluem análises históricas, como verificar se um caminhão visitou um determinado bairro durante a semana, identificar o dia e horário em que o caminhão esteve em um determinado bairro, além de verificar se o caminhão passou por uma rua específica durante um determinado período.

3.1.2.2 Aplicação 2: Monitoramento da Qualidade do Ar

A aplicação de monitoramento da qualidade do ar urbano tem como objetivo principal fornecer informações sobre a qualidade do ar e material particulado em áreas urbanas, permi-

tindo aos usuários monitorar e entender os níveis de poluição e outros parâmetros relacionados (PASTÓRIO *et al.*, 2022).

Os dados são captados por sensores instalados em estações de monitoramento distribuídas pela área urbana. Esses sensores medem parâmetros como dióxido de carbono (CO₂), dióxido de nitrogênio (NO₂), ozônio (O₃), monóxido de carbono (CO) e partículas suspensas no ar, como poeira e fumaça.

A aplicação disponibiliza dados através de uma base de dados ElasticSearch. E a partir desses dados o objetivo é fazer consultas como localização das estações de monitoramento em um mapa, filtrar os dados por intervalo de datas, visualizar histórico de medições, identificar os horários com melhores e piores índices e analisar os dias da semana com melhores e piores índices de qualidade do ar e material particulado.

Desta forma, ao utilizar as ferramentas selecionadas para comparação, é possível avaliar a eficácia da visualização na apresentação das informações desejadas.

3.2 Plataformas de Software para Cidades Inteligentes

As plataformas de software desempenham um papel fundamental, oferecendo ferramentas e recursos para coleta, análise e utilização de dados, bem como para a automação e otimização de processos. Esta seção aborda as plataformas de software utilizadas para o desenvolvimento deste trabalho.

3.2.1 Dados das aplicações

Conforme descrito na seções 3.1.2.2 e 3.1.2.1, uma aplicação de monitoramento dos caminhões de lixo e outra aplicação de monitoramento da qualidade do ar foram escolhidas como fontes de dados para as ferramentas Grafana e Kibana para o desenvolvimento deste estudo comparativo. O objetivo desta subseção é descrever a forma com que os dados serão entregues para o consumo das ferramentas de visualização.

Ambas aplicações estão hospedadas em um servidor na UTFPR campus Toledo e dispõem seus dados através de uma base de dados Elasticsearch. No entanto, é importante notar que a organização dos registros de eventos, conhecidos como *logs*, e dos atributos específicos que descrevem esses eventos, chamados de *fields* ou campos, difere entre as duas aplicações. Os *logs* representam registros detalhados das atividades de cada aplicação, enquanto os *fields* são os atributos que fornecem informações específicas sobre esses eventos. A variação na organização de *logs* e *fields* reflete as necessidades e a estrutura exclusiva de cada aplicação.

Na aplicação 1 (Monitoramento dos caminhões da coleta de lixo) os *logs* são gerados em curtos períodos e contém os seguintes campos com informações dos caminhões:

- Tempo: que representa o horário exato em que esse registro foi inserido no banco.

- Dispositivo: que representa o nome que foi dado ao dispositivo com GPS instalado no caminhão.
- Aplicação: que representa o nome dado a aplicação que coleta os dados e envia ao banco.
- Latitude: que representa a latitude da coordenada geográfica do caminhão.
- Longitude: que representa a longitude da coordenada geográfica do caminhão.

Para melhor representação de como os dados são disponibilizados, a Tabela 1 apresenta o resultado de uma consulta no Elasticsearch da aplicação em um período de alguns minutos.

Tabela 1 – Dados da aplicação 1

Tempo	Latitude	Longitude	Aplicação	Dispositivo
2023-12-02 05:28:43	-24.72235	-53.75665	toledo-rastreamento	tbeam-utfprtd-rastreamento-3
2023-12-02 05:28:00	-24.72234	-53.75665	toledo-rastreamento	tbeam-utfprtd-rastreamento-3
2023-12-02 05:27:23	-24.72235	-53.75665	toledo-rastreamento	tbeam-utfprtd-rastreamento-3
2023-12-02 05:26:33	-24.72238	-53.75663	toledo-rastreamento	tbeam-utfprtd-rastreamento-3
2023-12-02 05:25:49	-24.72236	-53.75663	toledo-rastreamento	tbeam-utfprtd-rastreamento-3
2023-12-02 05:25:13	-24.72235	-53.75663	toledo-rastreamento	tbeam-utfprtd-rastreamento-3
2023-12-02 05:24:22	-24.72238	-53.75663	toledo-rastreamento	tbeam-utfprtd-rastreamento-3
2023-12-02 05:23:39	-24.72237	-53.75663	toledo-rastreamento	tbeam-utfprtd-rastreamento-3

Fonte: Autoria Própria (2024).

Além dos campos acima citados, existem alguns outros campos são utilizados como forma de redundância e alguns para indexação dos dados na base de dados. Estes campos não foram utilizados na realização deste trabalho. Os campos disponíveis na base de dados são: @timestamp, @version, _id, _index, _type, altitude, coordinates.lat, coordinates.lon, hdop, highlight, location, provider, publishedAt, publisher, publisherTenant, sender, sensor, sort, speed, tenant, e type. Porém, utilizou-se apenas os campos: coordinates.lat, coordinates.lon, provider, sensor e @timestamp.

Assim como na aplicação 1, a aplicação 2 (Monitoramento da qualidade do ar) envia constantemente dados através de sua infraestrutura ao banco Elasticsearch. Os *logs* são gerados contendo os seguintes campos:

- Hora: que representa o horário exato em que este *log* foi inserido na base de dados.
- Publisher: que representa a estação de coleta que fez o envio destes dados.
- Tipo de Sensor: que representa o modelo do sensor que captou os dados.
- Nome de Sensor: que representa o nome dado ao sensor no momento da configuração da estação.
- Valor: que representa o valor captado pelo respectivo sensor no momento do registro.

Para entender melhor a real representação dos *logs* que são persistidos na base de dados Elasticsearch, a tabela 2 apresenta uma consulta de um pouco mais de 10 minutos diretamente no banco.

Tabela 2 – Dados da aplicação 2

Hora	Publisher	Tipo de Sensor	Nome de Sensor	Valor
Aug 28, 2023 @ 05:20:37.000	PM_station	MP_25	mp-proto-01_MP_25_1	9
Aug 28, 2023 @ 05:20:37.000	PM_station	UMIDADE	mp-proto-01_Umidade	86.8
Aug 28, 2023 @ 05:20:37.000	PM_station	TEMPERATURA	mp-proto-01_Temperatura	24.6
Aug 28, 2023 @ 05:09:50.000	PM_station	MP_25	mp-proto-01_MP_25_2	1
Aug 28, 2023 @ 05:09:50.000	PM_station	TEMPERATURA	mp-proto-01_Temperatura	24.6
Aug 28, 2023 @ 05:09:50.000	PM_station	UMIDADE	mp-proto-01_Umidade	86.9
Aug 28, 2023 @ 05:09:50.000	PM_station	MP_10	mp-proto-01_MP_10_1	8
Aug 28, 2023 @ 05:09:50.000	PM_station	MP_25	mp-proto-01_MP_25_1	7

Fonte: Autoria Própria (2023).

3.3 Métodos

Esta seção demonstra como as tecnologias citadas na seção 3.1 foram empregadas para realização do trabalho.

3.3.1 Infraestrutura Básica

Para cumprir os objetivos deste estudo comparativo entre Grafana e Kibana, foi necessário o processo de configuração da infraestrutura necessária. O foco inicial foi estabelecer os elementos essenciais para a realização da análise e comparação das duas ferramentas de visualização de dados.

1. **Acesso ao Servidor:** Inicialmente, obteve-se acesso ao servidor provido pela UTFPR Campus Toledo. Esse servidor serviu como o ambiente de hospedagem principal para as ferramentas Grafana e Kibana. O acesso foi concedido mediante autorização da universidade.
2. **Configuração Inicial:** Após obter acesso ao servidor, realizou-se uma configuração inicial para garantir que o servidor estava apto para prosseguir com as instalações. Isso incluiu a verificação da conectividade de rede, autenticação de usuário e a verificação de permissões necessárias.
3. **Instalação do Docker:** Uma etapa fundamental foi a instalação do Docker, uma plataforma de containerização. O Docker já estava previamente instalado no servidor, o que simplificou o processo. A capacidade de usar contêineres facilitou a implantação das ferramentas de análise e garantiu a consistência no ambiente de desenvolvimento.

Esses passos iniciais de configuração e instalação forneceram a base necessária para prosseguir com nossa análise comparativa entre Grafana e Kibana.

3.3.2 Instalação e Configuração do Grafana

A implementação do Grafana como parte deste estudo comparativo foi conduzida por meio da utilização do Docker, em conjunto com uma imagem oficial do Grafana disponível no Docker Hub, contendo a versão mais recente. Essa abordagem possibilitou uma instalação eficaz e replicável, mantendo a uniformidade no ambiente de pesquisa.

O Grafana foi instalado mediante a configuração do arquivo YAML do Docker, o qual foi utilizado para estabelecer os parâmetros necessários à execução do Grafana em um contêiner Docker. No código Docker mostrado na Listagem 3.1, foi definido a configuração de um contêiner para a execução do Grafana.

```
1 version: '3.8'
2 services:
3   grafana:
4     image: grafana/grafana
5     container_name: grafana
6     restart: unless-stopped
7     ports:
8       - '3000:3000'
9     volumes:
10      - grafana_data:/var/lib/grafana
11 volumes:
12   grafana_data: {}
```

Listing 3.1 – Arquivo yaml para executar o Grafana

Neste arquivo YAML, os principais pontos de configuração incluem:

- `image`: Especifica a imagem oficial do Grafana a ser utilizada.
- `container_name`: Define o nome do contêiner como "grafana".
- `restart`: Configurado para reiniciar o contêiner, a menos que a parada seja manual.
- `ports`: Mapeia a porta 3000 do host para a porta 3000 do contêiner, permitindo o acesso ao Grafana por meio do navegador.
- `volumes`: Define um volume chamado "grafana_data" para persistência dos dados, mapeando o diretório `/var/lib/grafana` dentro do contêiner.

Essa configuração resultou na criação de um ambiente de execução reproduzível para o Grafana, garantindo uma base para a análise comparativa.

Após a instalação do Grafana e com a aplicação em execução, o primeiro acesso é realizado utilizando as credenciais padrão de administrador, que são "admin:admin". Uma vez autenticado, é solicitado a modificação dessas credenciais para garantir a segurança do sistema. Com o acesso ao Grafana, é possível configurar as principais opções diretamente por meio da interface gráfica. O Grafana fornece uma interface intuitiva que permite personalizar e ajustar a plataforma de acordo com as necessidades específicas deste estudo comparativo.

A configuração que foi necessária é a conexão com a base de dados das aplicações de monitoramento do ar e monitoramento das coletas de lixo. Para isso utiliza-se a opção de configurar "*Data Sources*" que fica na aba de administração do Grafana.

Para configurar uma fonte de dados no Grafana, foram definidos diversos parâmetros. Inicialmente, especificou-se o tipo de banco de dados, que neste caso era o Elasticsearch, uma escolha para o acesso às informações de monitoramento do ar e dos caminhões de coletas de lixo. Além disso, foram fornecidos o endereço IP onde essa base de dados estava hospedada, garantindo a conectividade apropriada. As credenciais de autenticação também foram configuradas para assegurar o acesso autorizado ao banco de dados. Por fim, o nome do índice no Elasticsearch foi definido. O índice, neste contexto, representa a base de dados específica dentro do Elasticsearch, a partir da qual os dados seriam coletados para posterior análise (ELASTIC, 2021d).

Isso permitiu ao Grafana acessar e coletar informações diretamente dessas fontes de dados essenciais.

3.3.3 Instalação e Configuração do Kibana

A configuração do Kibana como parte deste estudo comparativo seguiu uma abordagem semelhante à utilizada para o Grafana. A instalação foi realizada por meio da plataforma Docker, com a escolha de uma imagem oficial do Kibana disponível no Docker Hub, assegurando a utilização da versão mais recente. Essa abordagem ofereceu uma instalação eficaz e replicável, mantendo a uniformidade do ambiente.

A implementação do Kibana foi facilitada pelo uso de um arquivo YAML do Docker, que definiu os parâmetros necessários para a execução do Kibana em um contêiner Docker. A seguir, apresenta-se o código Docker da Listagem 3.2, que descreve a configuração do contêiner para a execução do Kibana.

```

1 version: '2'
2 services:
3   kibana:
4     image: docker.elastic.co/kibana/kibana:8.10.4

```

```

5     container_name: kibana
6     restart: unless-stopped
7     ports:
8         - '5601:5601'
9     volumes:
10        - ./kibana.yml:/usr/share/kibana/config/kibana.yml

```

Listing 3.2 – Arquivo YAML para executar o Kibana

Neste arquivo YAML, as configurações incluem:

- `image`: Especificação da imagem oficial do Kibana, garantindo o uso da versão mais recente.
- `container_name`: Definição do nome do contêiner como "kibana".
- `restart`: Configuração para reiniciar o contêiner, a menos que a parada seja manual.
- `ports`: Mapeamento da porta 5601 do host para a porta 5601 do contêiner, permitindo o acesso ao Kibana por meio do navegador.

Essa configuração estabeleceu um ambiente de execução para o Kibana, fornecendo uma base para a análise comparativa.

O Kibana, assim como o Grafana, demanda configurações específicas para operar eficazmente no contexto deste estudo comparativo de cidades inteligentes. Uma vez que o contêiner do Kibana está em execução, a configuração pode ser realizada diretamente por meio da interface gráfica, acessível através do servidor web disponível na porta 5601 do host onde o Kibana está hospedado.

O Kibana está intrinsecamente relacionado ao Elasticsearch, e, portanto, a primeira etapa de configuração ao acessar a interface web do Kibana é a conexão com a base de dados Elasticsearch. Neste cenário, onde as bases de dados das aplicações de monitoramento do ar e monitoramento das coletas de lixo estão hospedadas no mesmo servidor onde o Kibana está em execução, o Kibana é capaz de identificar automaticamente a base de dados, realizando a conexão através do endereço 'localhost:9200', que é a porta padrão do Elasticsearch.

Essas configurações permitiram ao Kibana fornecer uma plataforma para a análise de dados neste estudo comparativo. Nas próximas seções, explora-se em detalhes o uso dessas configurações para a análise dos dados coletados.

3.3.4 Requisitos da Aplicação 1: Monitoramento da Coleta de Lixo

Para a implementação e avaliação do sistema de Monitoramento da Coleta de Lixo, foram utilizados requisitos específicos que visam garantir a eficácia e a funcionalidade da apli-

Tabela 3 – Requisitos da Aplicação de Monitoramento da Coleta de Lixo

ID	Descrição do Requisito
RF01	Apresentar um mapa da cidade com todos os veículos
RF02	Apresentar a localização do usuário no mapa
RF03	Apresentar a localização de veículo específico no mapa
RF04	Apresentar o histórico de rotas dos veículos
RF05	Apresentar o histórico de coleta/passagem do caminhão em determinada localização

(PAGANINI, 2024)

cação. Esses requisitos foram definidos no trabalho de mestrado que esta em desenvolvimento por Paganini (2024).

Os requisitos da aplicação, listados na Tabela 3, abrangem desde a visualização da frota de veículos até o acompanhamento do histórico de rotas e coletas. Cada requisito foi selecionado para atender às demandas específicas do monitoramento da coleta de lixo, proporcionando uma visão abrangente das funcionalidades.

3.3.5 Requisitos da aplicação 2: Monitoramento da Qualidade do Ar

Para a comparação das ferramentas Grafana e Kibana foram utilizados como parâmetros os requisitos de aplicação definidos no trabalho de mestrado que está sendo desenvolvido pelo Paganini (2024).

Os requisitos da aplicação, listados na Tabela 4, foram selecionados para abranger os aspectos fundamentais do sistema de monitoramento da qualidade do ar. Eles abrangem desde a coleta e armazenamento de dados até a apresentação de informações relevantes aos usuários.

Tabela 4 – Requisitos da aplicação de monitoramento da qualidade do ar

ID	Descrição do Requisito
RF01	Armazenar as medições dos sensores das estações
RF02	Apresentar um mapa da cidade com todas as estações
RF03	Apresentar as medições dos sensores das estações
RF04	Apresentar o histórico de medições dos sensores das estações
RF05	Apresentar o Índice de qualidade do Ar (IQA) das estações
RF06	Apresentar o histórico do Índice de qualidade do Ar (IQA) das estações
RF07	Apresentar o Índice de calor das estações
RF08	Apresentar o histórico do Índice de calor das estações
RF09	Apresentar a localização do usuário no mapa
RF10	Apresentar os melhores e piores índices e parâmetros das últimas 24 horas
RF11	Apresentar o histórico dos melhores e piores índices e parâmetros
RF12	Apresentar o dia da semana com os melhores e piores índices e parâmetros
RF13	Apresentar o horário com os melhores e piores índices e parâmetros para cada estação

(PAGANINI, 2024)

Com os requisitos estabelecidos, para entender se um requisito foi ou não atendido deve se levar em conta a capacidade de um usuário entender as informações apresentadas, e além disso, ser capaz de conseguir acessar as informações de acordo com os parâmetros que deseja.

4 RESULTADOS

Este Capítulo apresenta o que foi obtido como resultado deste trabalho comparativo entre as ferramentas Grafana e Kibana. São apresentados o processo de criação dos *dashboard* e quais comportamentos obtivemos a partir dos requisitos definidos no trabalho de mestrado do Paganini (2024).

4.1 Criação dos *dashboards* no Grafana

Para atingir o principal objetivo deste estudo comparativo, a montagem dos painéis de visualização são fundamentais para compreender as dificuldades e facilidades de cada ferramenta. Abaixo estão descritos os caminhos para a construção dos *dashboards* no Grafana para as aplicações de monitoramento dos caminhões de lixo (aplicação 1) e da qualidade do ar (aplicação 2).

4.1.1 Painéis da aplicação 1 no Grafana

Após a coleta dos dados realizada conforme demonstrado na Tabela 1, foi possível realizar o tratamento dos dados para a elaboração do *dashboard* para a visualização dos dados e posterior análise dos resultados. A Figura 2 é a representação da página do *dashboard*, cujo apresenta o mapa de rastreamento dos caminhões da coleta de lixo.

Para a criação desse painel, foi utilizado apenas um tipo de painel, chamado Geomap, que faz parte dos recursos padrões do Grafana. Esse tipo de visualização permite plotar coordenadas geográficas em um mapa a partir de diferentes fontes de dados. Como o objetivo desse painel era apresentar a localização dos caminhões da coleta de lixo e atender aos requisitos impostos pela aplicação, foram criadas consultas separadas para cada caminhão, garantindo que o Geomap plotasse corretamente as informações.

A Figura 3 demonstra como foi realizada a configuração para a consulta retornar os dados do caminhão especificado - neste caso do caminhão 3. O campo 'Query Type' indica o tipo de retorno que o usuário deseja obter da consulta, nesta consulta utilizou-se o tipo 'Raw Data' (em português: dado cru), ou seja irá retornar os dados exatamente do jeito que a base de dados os armazenou. Como até esse ponto a consulta não tinha nenhum tipo de filtragem o retorno iria ser vários registros de todos os caminhões existentes no banco.

Para filtrar os dados e obter somente o caminhão desejado foi necessário configurar o campo 'Lucene Query', esse campo é responsável por fazer a filtragem dos dados com a escrita de uma *query* utilizando a linguagem Lucene, que é uma linguagem específica para bases ElasticSearch. Neste campo utilizamos a seguinte *query*: 'sensor : "tbeam-utfprtd-rastreamento-3"'. Esta busca especifica que o campo 'sensor' tem que ser igual a 'tbeam-utfprtd-rastreamento-

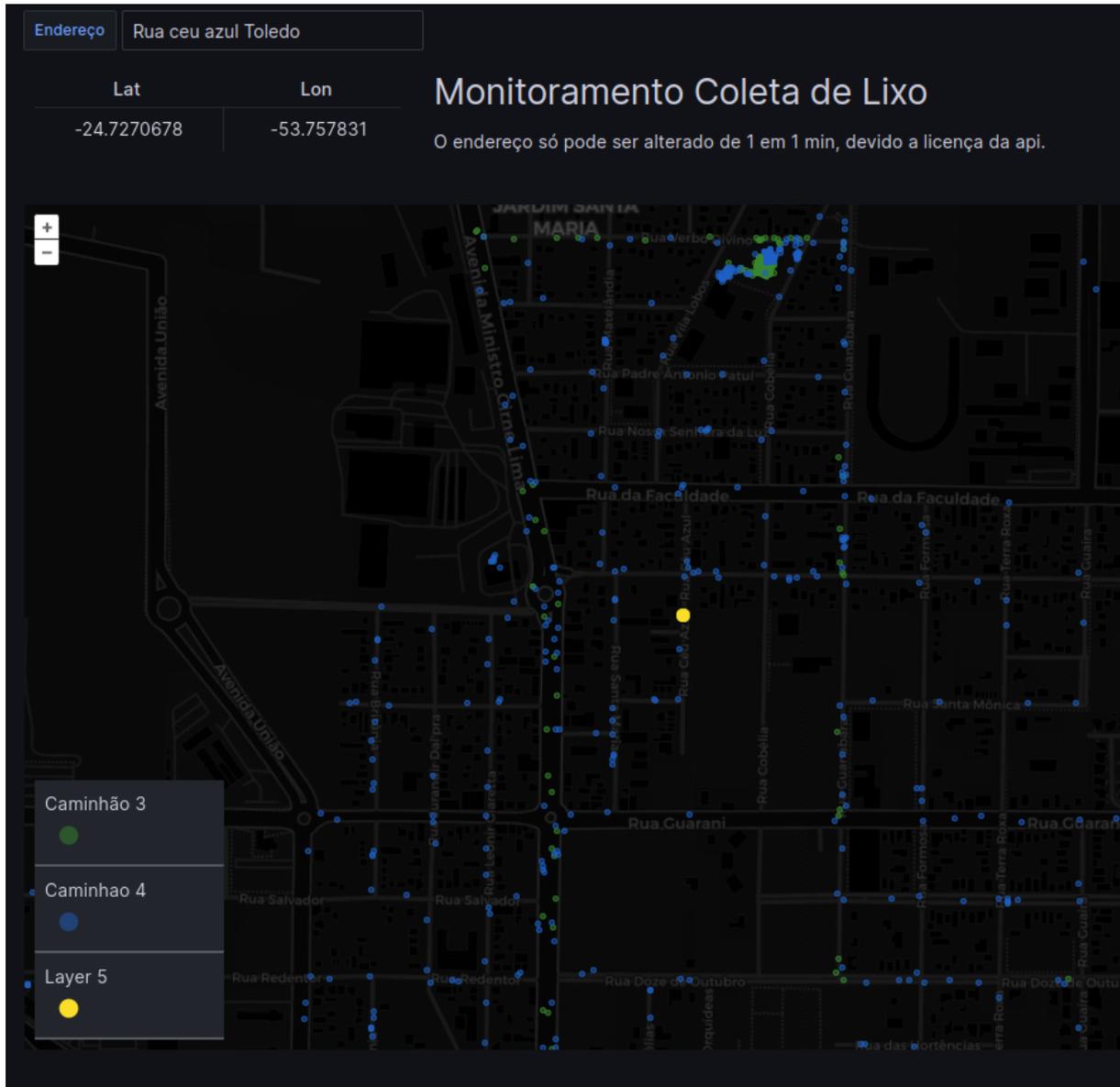


Figura 2 – Aplicação 1

Fonte: Autoria Própria

3', que é a identificação do caminhão 3, desta forma retornando apenas registros do caminhão 3.

Essa consulta foi replicada para todos os caminhões que possuíam dispositivos de rastreamento, alterando apenas o filtro para cada identificação de caminhão na base de dados.

Neste *dashboard* foi adicionado um ponto de centralização do mapa, permitindo que o usuário digite um endereço e o mapa se centralize nas coordenadas desse endereço. Para isso foi utilizada uma API externa que retorna coordenadas geográficas a partir de um endereço. A API utilizada se chama 'GeoCode' e para utilizá-la basta fazer uma requisição HTTP do tipo GET no endereço web 'https://geocode.maps.co/search' passando o endereço desejado.

A configuração dos parâmetros da requisição HTTP é demonstrada na Figura 4. Na aba 'Field', foram configurados os parâmetros de retorno da API, sabendo que ela retornaria

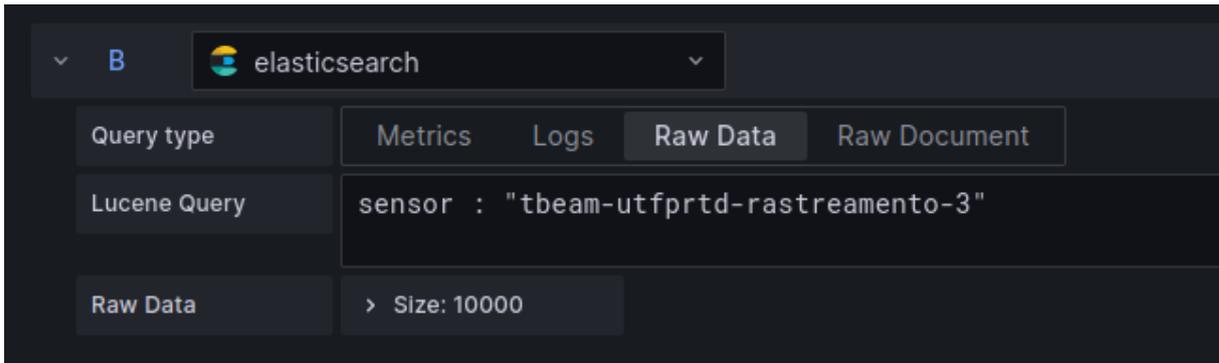


Figura 3 – Consulta Caminhão Grafana
Fonte: Autoria Própria

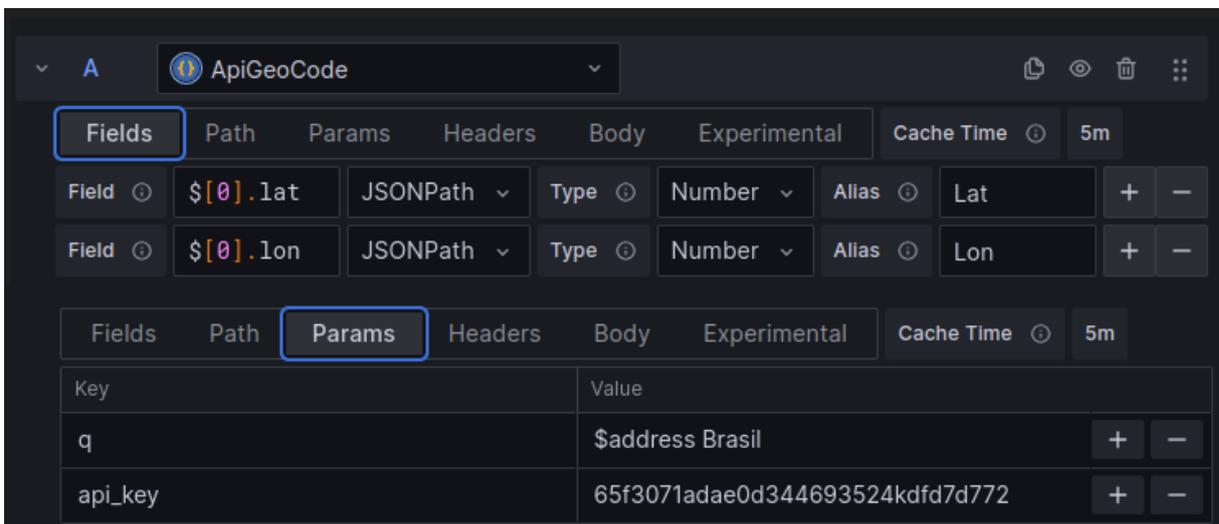


Figura 4 – Configuração API
Fonte: Autoria Própria

um JSON com os campos 'lat' e 'lon', representando as coordenadas. Na aba 'Params', foram configurados os parâmetros de busca da requisição, incluindo uma chave da API e o endereço digitado pelo usuário. Dessa forma, foram obtidas as coordenadas geográficas do ponto especificado pelo usuário, representado na cor amarela, conforme mostrado na Figura 2.

4.1.2 Painéis da aplicação 2 no Grafana

Após a coleta dos dados realizada conforme demonstrado na Tabela 2, os dados foram processados para a criação do *dashboard*, permitindo a visualização e análise dos resultados. A Figura 5 mostra a página do *dashboard*, apresentando os painéis criados para a visualização das métricas da estação de material particulado.

Para o desenvolvimento deste *dashboard* foram utilizados alguns tipos diferentes de visualizações de acordo com a métrica a se visualizar, tornando sua compreensão mais fácil. Para o sensor MP-10 foi utilizado o tipo de visualização 'Time Series' presente nos recursos



Figura 5 – Aplicação 2

Fonte: Autoria Própria

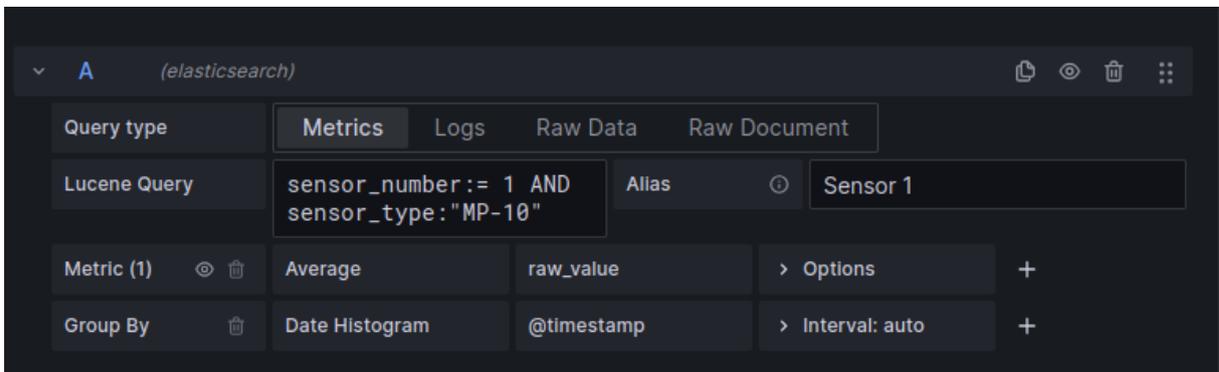


Figura 6 – Configuração Sensor MP10

Fonte: Autoria Própria

padrões do Grafana. Como o próprio nome diz, esse tipo de visualização cria uma serie temporal cruzando os dados desejados como um campo de tempo.

A Figura 6 demonstra as configurações realizadas e as consultas utilizadas para fazer a plotagem dos dados no gráfico. Neste caso o objetivo era plotar os dados de dois sensores na mesma serie temporal, para isso precisou-se criar as consultas de busca na base de dados ElasticSearch visando obter somente os registros dos sensores desejados.

O campo 'Query Type' indica o tipo de retorno que o usuário deseja obter da consulta, nesta consulta utilizou-se o tipo 'Metrics', neste método o Grafana realiza uma varredura por todos os campos retornados da busca no banco e cria uma caixa de seleção com todos os campos encontrados, bastando apenas selecionar o campo desejado para a plotagem. Como até esse ponto não tinha nenhum tipo de filtragem o retorno iria ser vários registros de todos os sensores existentes no banco.

Para a filtragem apenas dos sensores desejados precisamos utilizar uma Lucene Query, assim como no *dashboard* da aplicação 1. Neste campo utilizamos a seguinte *query*: 'sensor_number:= 1 AND sensor_type:"MP-10"'. Esta busca especifica que o campo 'sensor_number'

tem que ser igual a 1, que é o número do sensor na base de dados e o campo 'sensor_type' deve ser igual a "MP-10", que é o tipo de sensor que desejamos buscar, desta forma retornando apenas registros do sensor desejado.

Como esse painel era responsável pela plotagem de dois sensores, foi necessário replicar essa consulta para o outro sensor, mudando apenas o número do sensor especificado na Lucene Query da busca. Assim, a Figura 7 representa a plotagem da serie temporal com os dados dos sensores em cores diferentes.

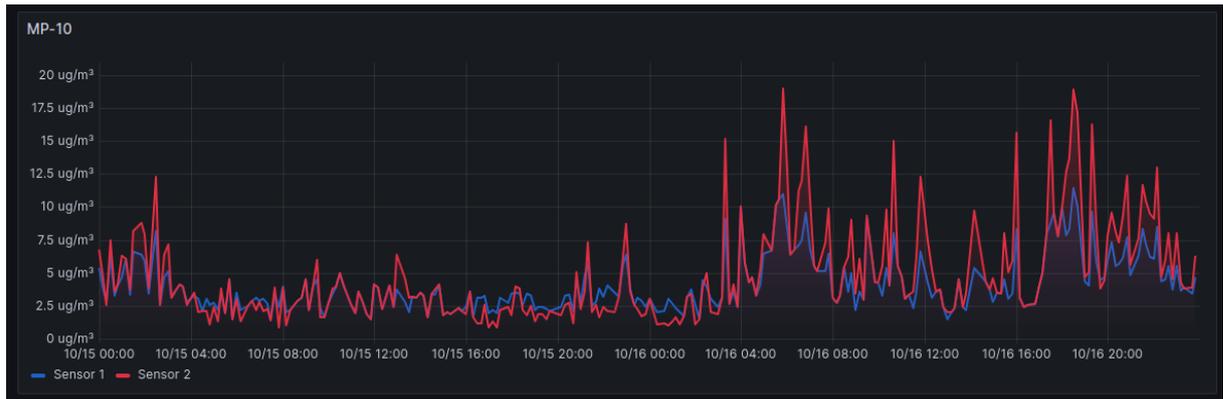


Figura 7 – Painel MP10

Fonte: Autoria Própria

O tipo de painel 'Time Series' também foi utilizado para a visualização dos sensores de temperatura, umidade e o sensor de material particulado MP2.5. A Figura 8 apresenta a plotagem dos gráficos a partir das consultas realizadas na base de dados da aplicação. As configurações foram semelhantes as anteriormente apresentadas, alterando sempre a Query escrita para a busca dos sensores corretos.



Figura 8 – Painéis de série temporal

Fonte: Autoria Própria

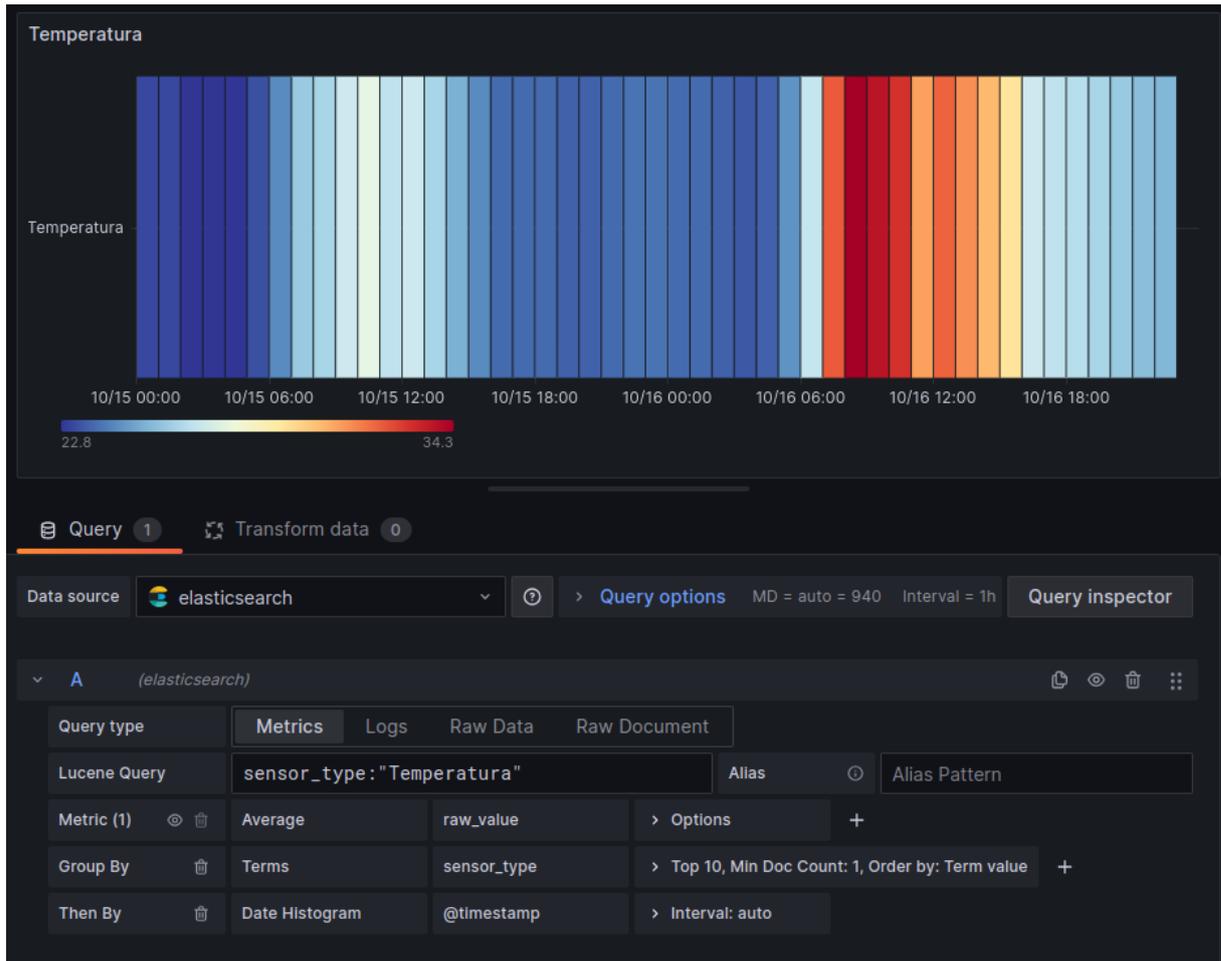


Figura 9 – Mapa de Calor Temperatura

Fonte: Autoria Própria

Outro tipo de painel utilizado é o Mapa de Calor, um painel que plota um mapa de calor de acordo com a métrica selecionada. Utilizou-se este tipo de visualização para visualizar de maneira diferente o comportamento da temperatura durante o período desejado. A Figura 9 apresenta a configuração e a plotagem resultante desta configuração para esse painel.

A configuração deste tipo de painel é muito semelhante ao de Serie Temporal, o único ponto diferente a se configurar é a paleta de cores utilizada para a plotagem do mapa de calor. A Figura 10 demonstra a configuração das cores utilizadas. Para essa configuração foi utilizada a escala automática do Grafana, cujo a ferramenta busca o mínimo e máximo valor e divide a paleta de cores entre elas, tornando a plotagem e a transição das cores perfeitas em relação aos dados.

O terceiro e último tipo de painel utilizado na construção do *dashboard* da aplicação 2 foi o 'Gauge', uma visualização que apresenta os valores como se fosse um visor ou medidor. Este é o tipo de visualização mais simples que o Grafana oferece, basta configurar os parâmetros de busca e então o campo desejado será apresentado. Foi utilizado esse tipo de visualização para demonstrar os dados de três sensores diferentes, o de temperatura, o de umidade e o sensor que mede a velocidade dos ventos, o anemômetro.

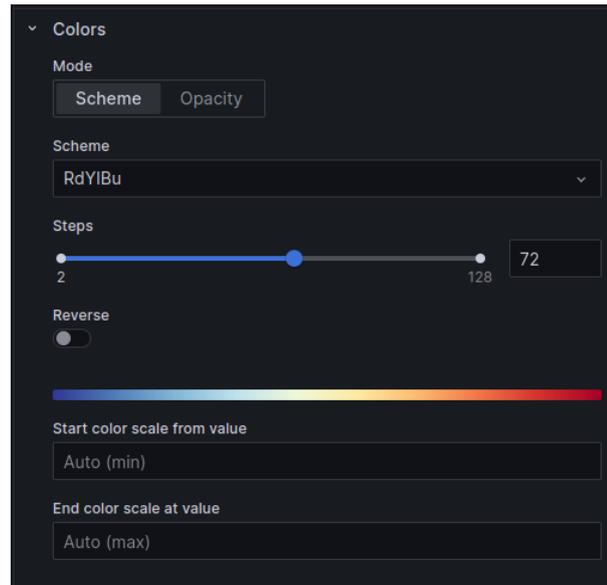


Figura 10 – Mapa de Calor Cores
Fonte: Autoria Própria

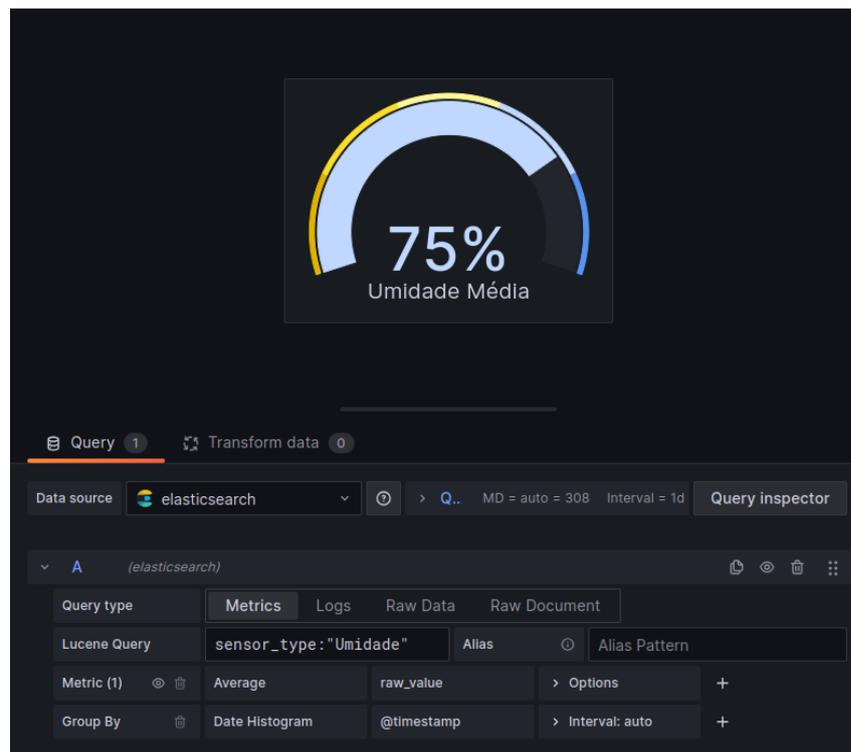


Figura 11 – Anemômetro
Fonte: Autoria Própria

A Figura 11 apresenta as configurações de busca e a visualização da métrica selecionada representada em um painel do tipo Gauge. As configurações de busca são semelhantes aos outros painéis tendo apenas que filtrar os valores pela Lucene Query, e então selecionar o campo a ser apresentado.

Um ponto interessante desse tipo de visualização é que o usuário pode escolher a forma com que os valores serão agrupados, ou então qual valor será mostrado. Por exemplo, neste

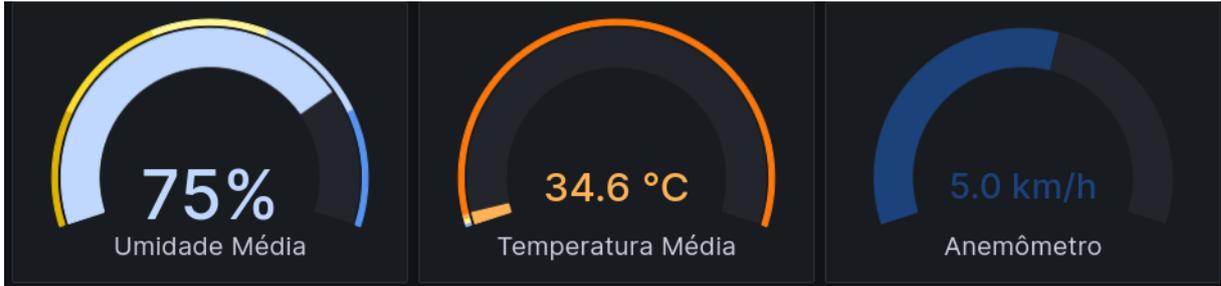


Figura 12 – Painéis Gauge

Fonte: Autoria Própria

trabalho utilizou-se a média dos valores do período para se mostrar, mas este poderia ser o primeiro valor encontrado ou o ultimo valor. Isso pode ser configurado no campo 'Metric' no Grafana.

A Figura 12 apresenta os 3 painéis que utilizaram a visualização do tipo Gauge para apresentar as métricas.

4.2 Criação dos *dashboards* no Kibana

Para atingir o principal objetivo deste estudo comparativo, a montagem dos painéis de visualização são fundamentais para compreender as dificuldades e facilidades de cada ferramenta. Abaixo estão descritos os caminhos para a construção dos *dashboards* na ferramenta Kibana para as aplicações de monitoramento dos caminhões de lixo (aplicação 1) e da qualidade do ar (aplicação 2).

4.2.1 Painéis da aplicação 1 no Kibana

Após a coleta dos dados realizada conforme demonstrado na Tabela 1, foi possível realizar a elaboração do *dashboard* no Kibana para a visualização dos dados e posterior análise dos resultados. A Figura 13 é a representação da página do *dashboard*, onde contém o mapa de rastreamento dos caminhões da coleta de lixo.

Para a criação desse painel, foi utilizado apenas um tipo de painel, chamado Maps, que faz parte dos recursos padrões do Kibana. Esse tipo de visualização permite plotar coordenadas geográficas em um mapa a partir dos dados presentes na base de dados Elasticsearch. Como o objetivo desse painel era apresentar a localização dos caminhões da coleta de lixo, foram criadas consultas separadas para cada caminhão, garantindo que o Kibana plotasse corretamente as informações.

Diferentemente do Grafana as consultas no Kibana são mais simples de se configurar, uma vez que o Kibana trabalha em conjunto com o Elasticsearch. A alta compatibilidade entre as duas ferramentas permite que com poucos parâmetros o painel Maps já reconheça as coordenadas e plote-as no mapa. Contudo, mesmo com a compatibilidade citada, como o objetivo

Monitoramento Coleta de Lixo

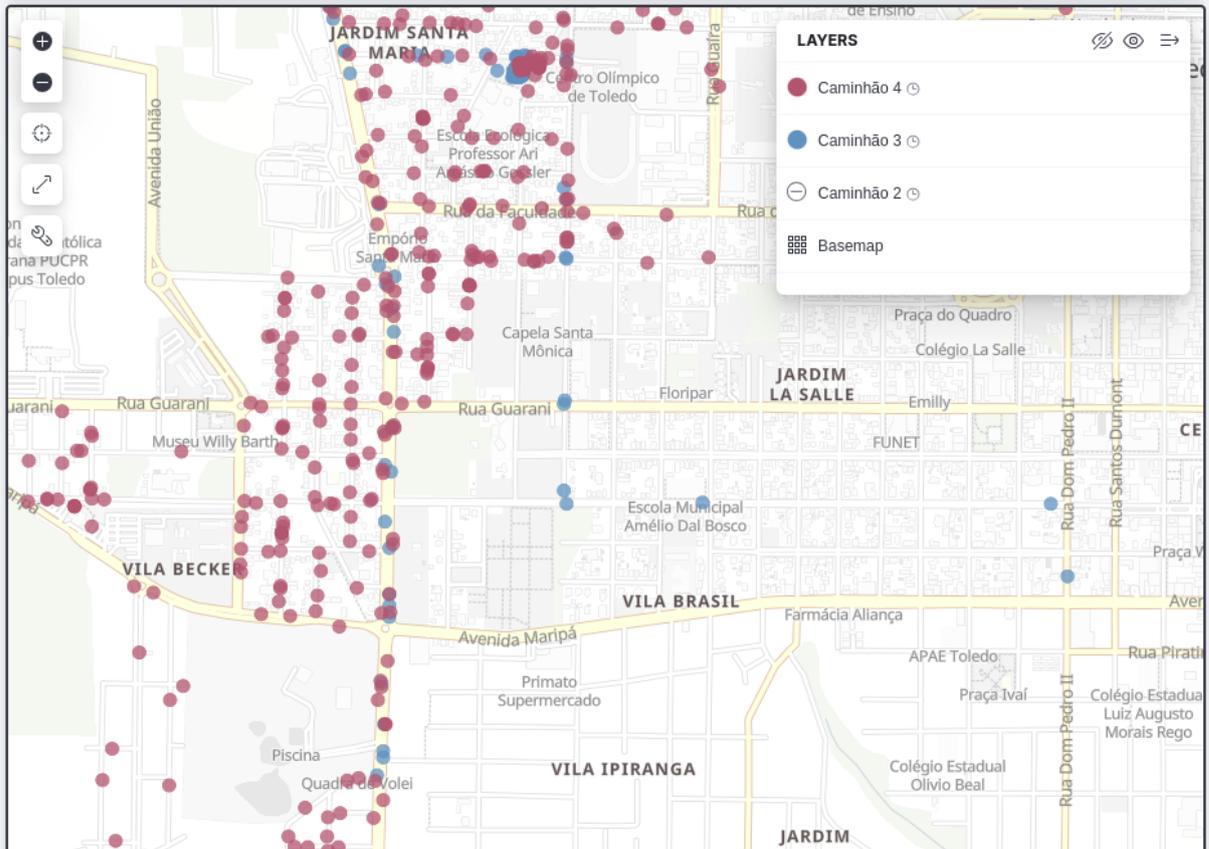


Figura 13 – Aplicação 1 - Kibana

Fonte: Autoria Própria

desse painel era apresentar a localização dos caminhões da coleta de lixo, foram criadas consultas separadas para cada caminhão visando separar os pontos de cada caminhão com cores distintas.

A Figura 14 demonstra como foi realizada a configuração para a consulta retornar os dados do caminhão especificado - neste caso do caminhão 3. Na parte superior se encontra a aba de configuração da fonte de dados. Nesta aba o Kibana já identifica automaticamente a presença do campo 'coordinates' no banco de dados, e seleciona este como fonte para a plotagem das coordenadas geográficas. O campo 'coordinates' possui em seus registros as latitudes e longitudes de cada caminhão equipado com dispositivo de rastreamento.

Como até esse ponto não havia nenhum tipo de filtragem, o retorno seria registros de todos os caminhões existentes no banco. Para filtrar os dados e obter somente registros do caminhão desejado, foi necessário configurar o campo 'Filtering'. Esse campo espera como entrada uma Query escrita com a linguagem Lucene Query, que é nativa do Elasticsearch e específica pra filtragem dos dados. Neste campo utilizamos a seguinte *query*: 'sensor : "tbeam-utfprtd-rastreamento-3"'. Esta busca específica que o campo 'sensor' tem que ser igual a 'tbeam-

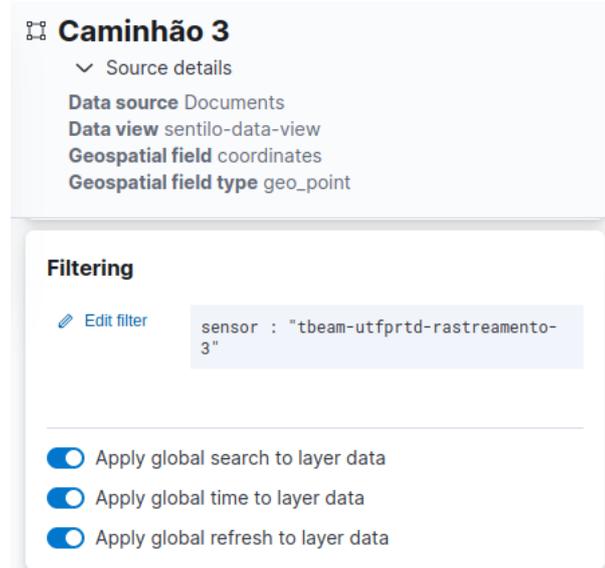


Figura 14 – Consulta Caminhão Kibana

Fonte: Autoria Própria

utfprtd-rastreamento-3', que é a identificação do caminhão 3, desta forma retornando apenas registros do caminhão 3.

Essa consulta foi replicada para todos os caminhões que possuíam dispositivos de rastreamento, alterando apenas o filtro para cada identificação de caminhão na base de dados.

No Kibana não foi possível a criação de um ponto de centralização do mapa, pois a ferramenta não suporta integrações com serviços externos como uma API. Desta forma, a centralização e posicionamento do mapa fica por conta do usuário.

4.2.2 Painéis da aplicação 2 no Kibana

Após a coleta dos dados realizada conforme demonstrado na Tabela 2, foi possível realizar elaboração do *dashboard* na plataforma Kibana para a visualização dos dados e posterior análise dos resultados. A Figura 15 é a representação da página do *dashboard*, cujo contem os painéis desenvolvidos para a visualização das métricas da estação de material particulado.

Para o desenvolvimento deste *dashboard* foram utilizados alguns tipos diferentes de visualizações de acordo com a métrica a se visualizar, tornando sua compreensão mais fácil. Para o sensor MP-10 foi utilizado o tipo de visualização 'Line' presente nos recursos padrões da ferramenta Kibana. Esse tipo de visualização cria um gráfico de linha cruzando os campos desejados com um campo de tempo. Esse painel é se assemelha ao padrão 'Time Series' do Grafana.

A Figura 16 demonstra as configurações iniciais realizadas para a plotagem do gráfico. O primeiro campo a se configurar é a fonte de dados, que nesse caso é a que se encontra os dados vindos da plataforma Sentilo. O próximo campo é a configuração do eixo horizontal, esse campo espera como parâmetro um campo de tempo para fazer a plotagem dos dados. O

Estação Material Particulado

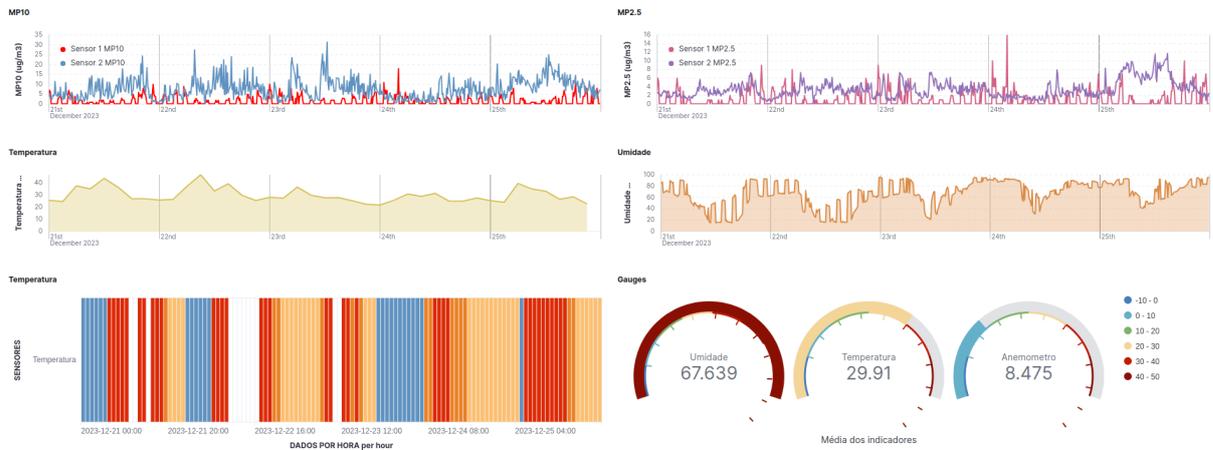


Figura 15 – Aplicação 2 - Kibana

Fonte: Autoria Própria

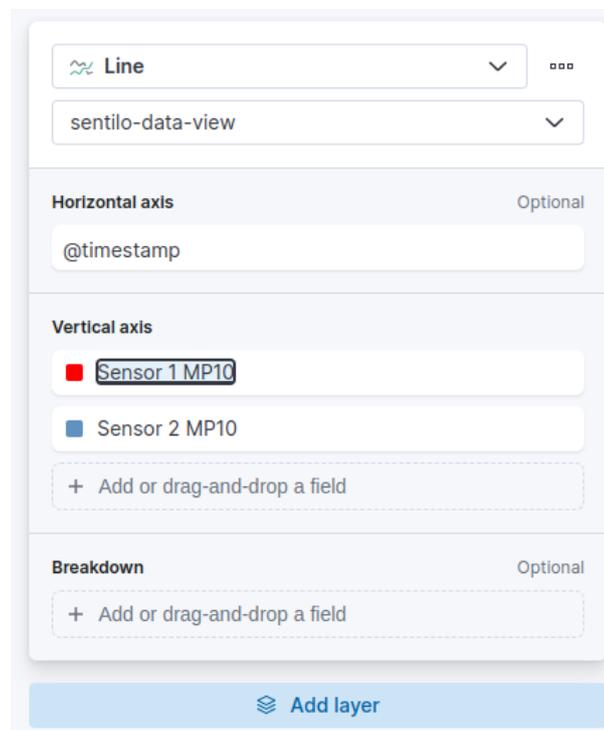


Figura 16 – Configuração Painel Line

Fonte: Autoria Própria

terceiro e ultimo campo é o de eixos verticais, neste painel foi selecionado dois sensores como fonte de dados.

Cada métrica selecionada como eixo vertical possui suas próprias configurações de filtragem e agregação dos dados. A Figura 17 demonstra como foi realizada a configuração de um eixo vertical para a apresentação dos dados de um sensor de material particulado. Existem duas formas de filtragem de dados definidos no campo 'Method', uma é 'Quick function' onde o usuário pode definir os parâmetros de filtragem diretamente pela interface, e o outro é 'For-



Vertical axis ✕

Method

Quick function **Formula**

Formula

☰ Expand

```
last_value(raw_value, kql='sensor_number: "1"')
```

📄

[> Advanced](#)

Appearance

Name

Value format

Series color 🎨

Axis side Left Auto Right

Figura 17 – Configuração Eixo Vertical
Fonte: Autoria Própria

mula' cujo o usuário escreve uma consulta utilizando a linguagem de busca KQL(Kibana Query Language) combinada com uma Query do tipo Lucene.

Neste painel utilizamos o método de filtragem 'Formula', configurando ele com a Query: "last_value(raw_value, kql='sensor_number: "1"")". A função 'last_value' indica que o usuário deseja obter o ultimo valor de um conjunto de dados, esse conjunto é definido dentro dos parênteses após a chamada da função. O primeiro parâmetro que é utilizado nessa função é o campo que deseja-se obter dentro do conjunto de dados, e o segundo parametro é uma Query de filtragem. Neste caso, a Query de filtragem instrui que os dados tenham o campo 'sensor_number' igual a 1.

Outra configuração realizada é a aparência das linhas no gráfico, onde é possível definir nomes personalizados para cada eixo vertical, o sentido e a cores das linhas.

Esse procedimento de configuração dos eixos verticais se repetem pra quantos eixos o usuário deseja inserir, alterando apenas o campo de busca e o tipo de agregação nos dados. Desta maneira, a Figura 18 demonstra a plotagem final do gráfico com a configuração realizada para dois sensores de material particulado do tipo MP10.

Outro tipo de painel utilizado é o Mapa de Calor, um painel que plota um mapa de calor de acordo com a métrica selecionada. Utilizou-se este tipo de visualização para visualizar de maneira diferente o comportamento da temperatura durante o período desejado. A Figura 19 apresenta a configuração utilizada para plotar os dados do sensor de temperatura.

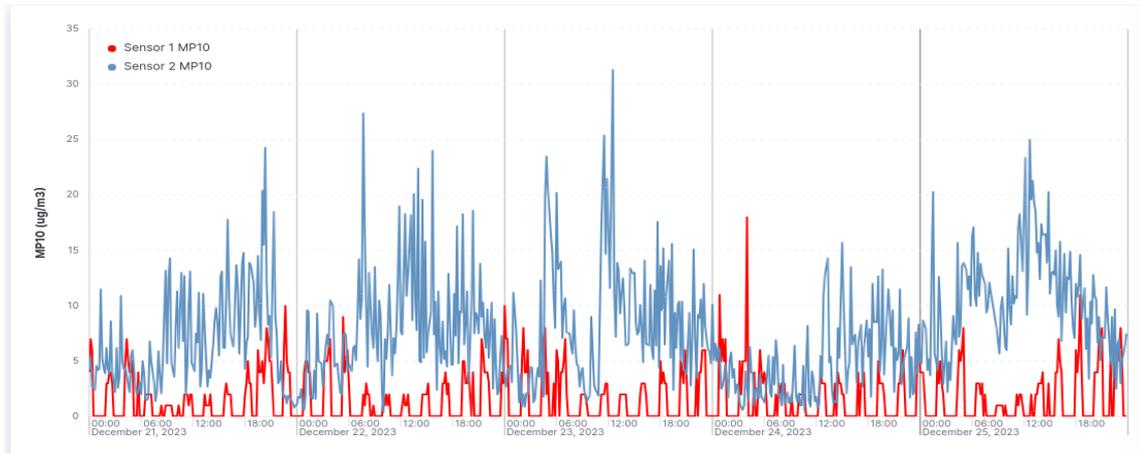


Figura 18 – Painel Line
Fonte: Autoria Própria

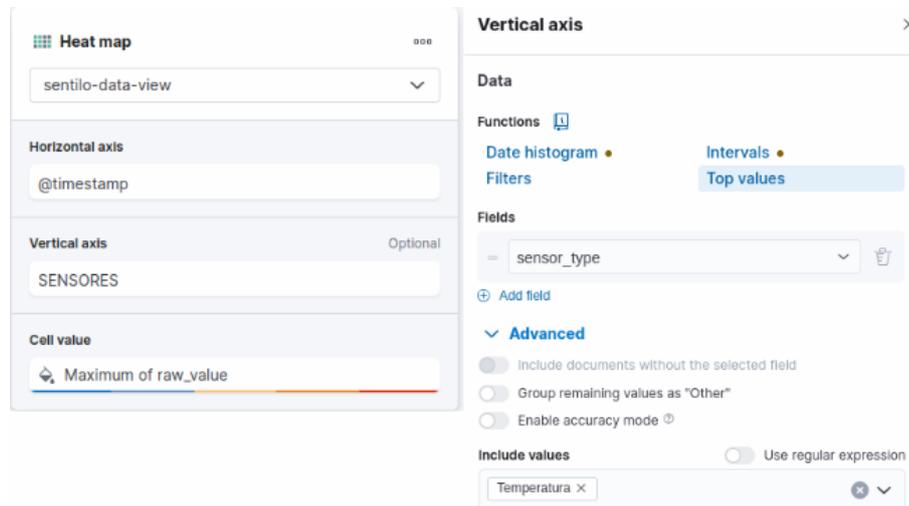


Figura 19 – Configuração Mapa de Calor
Fonte: Autoria Própria

A configuração se assemelha com a configuração dos outros tipos de painéis, porém, utilizou-se uma forma diferente de filtragem, selecionando o campo desejado e ao invés de escrever a Query para filtrar, faz a inclusão do valor desejado na aba 'Include Values'. Os valores já são apresentados automaticamente pelo Kibana ao selecionar o campo a se apresentar, então basta selecionar pelo valor que deseja filtrar - neste caso utilizamos o valor 'Temperatura' para obter apenas os dados do sensor de temperatura da estação de coleta.

Para definir as cores do Mapa de Calor foi selecionado uma paleta de cores e o próprio Kibana faz a escala automática dos valores para dividir a paleta de cores e aplicar a cor correta a depender do valor obtido da base de dados. A Figura 20 apresenta a plotagem do mapa de calor utilizando as configurações e parâmetros citados.

O terceiro e último tipo de painel utilizado na construção do *dashboard* da aplicação 2 no Kibana foi o 'Gauge', uma visualização que apresenta os valores na forma de um visor ou medidor. Para a utilização deste tipo de painel, basta configurar os parâmetros de busca e então

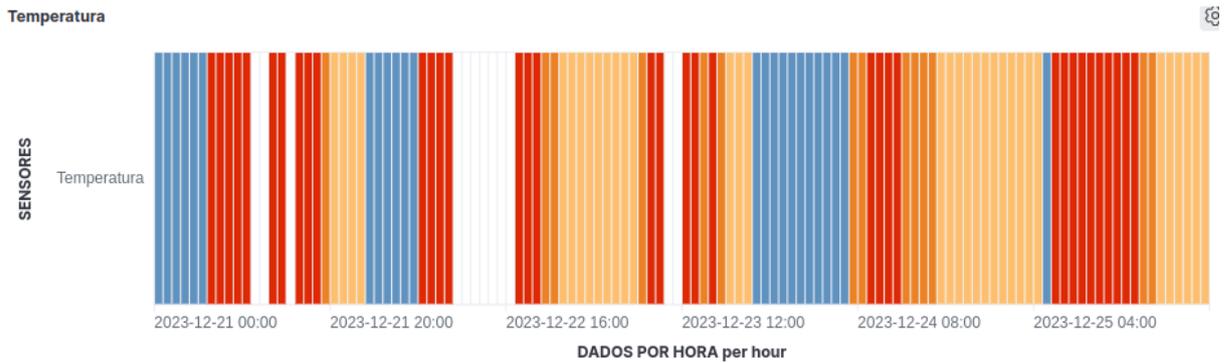


Figura 20 – Mapa de Calor Kibana

Fonte: Autoria Própria

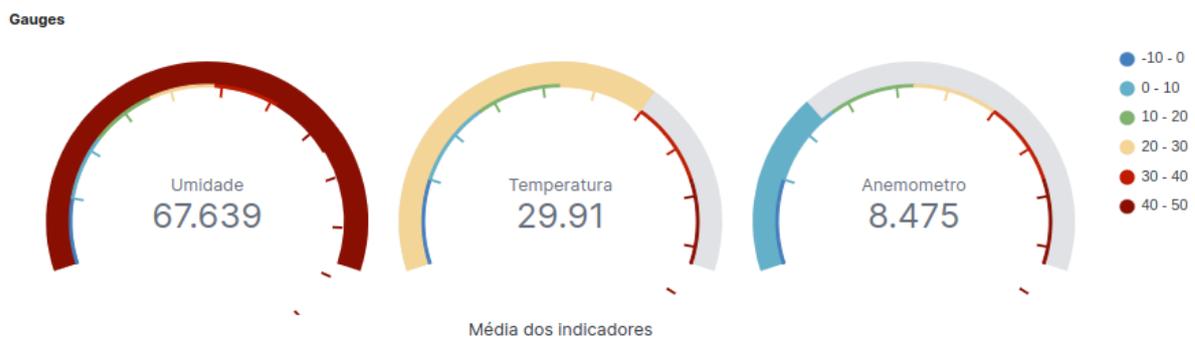


Figura 21 – Gauges Kibana

Fonte: Autoria Própria

o campo desejado será apresentado. Foi utilizado esse tipo de visualização para demonstrar os dados de três sensores diferentes, o de temperatura, o de umidade e o sensor que mede a velocidade dos ventos, o anemômetro, conforme demonstrado na Figura 21.

As configurações de busca são semelhantes aos outros painéis tendo apenas que filtrar os valores pela Query digitando o nome do campo a ser apresentado. A Figura 22 apresenta as configurações de busca e a visualização da métrica selecionada representada em um painel do tipo Gauge.

4.3 Comparação com Requisitos Definidos para Aplicação 1

Esta seção apresenta os resultados obtidos a partir dos requisitos definidos. Para cada requisito foi feita uma análise comparativa nas duas ferramentas utilizando os painéis de visualização criados e os recursos disponíveis em cada uma das ferramentas Grafana e Kibana.

4.3.1 RF01: Armazenar as medições dos sensores das estações

Esse requisito não envolve visualização de dados ou algo do tipo, mas sim a configuração para que os dados fossem armazenados e requisitados quando necessário. Como ambas

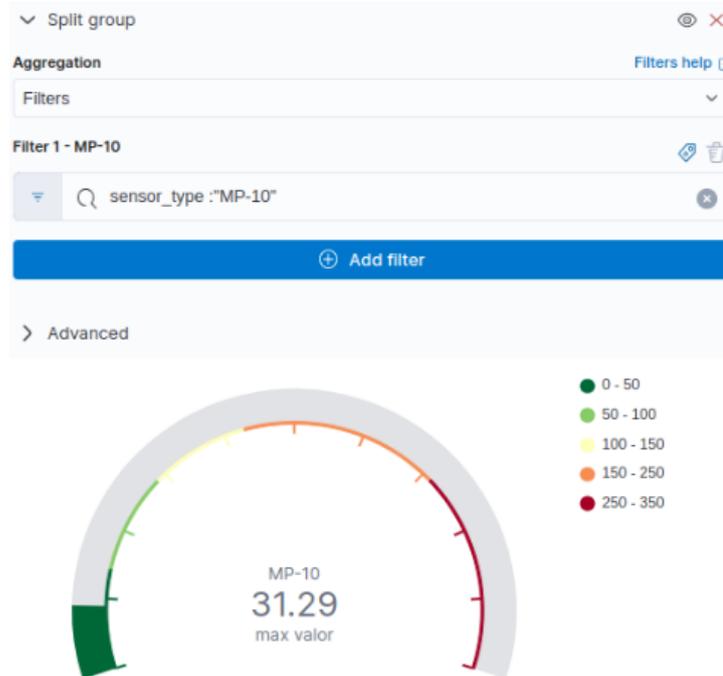


Figura 22 – Configuração Gauge Kibana
Fonte: Autoria Própria

ferramentas utilizavam a mesma base de dados Elasticsearch, atender à esse requisito ficou por conta da base de dados e não das ferramentas Grafana e Kibana. Apesar de tal constatação, podemos englobar a configuração das ferramentas para o busca dos dados no banco onde estão armazenados.

Levando a configuração da busca dos dados em consideração, o Kibana tem certa facilidade em lidar com base de dados Elasticsearch, já que foi desenvolvida para isto, enquanto no Grafana, é necessário configurar a conexão e autenticação conforme demonstrado na Subseção 3.3.2. Essa configuração é simples e semelhante a maioria dos tipos conexões a banco de dados existentes no mercado.

4.3.2 RF02: Apresentar um mapa da cidade com todas as estações

Esse requisito foi atendido nas duas ferramentas de visualização de dados de maneira muito semelhante. A configuração para apresentação envolveu apenas a seleção das coordenadas da estação nos campos do banco de dados.

Como na época do desenvolvimento deste trabalho só havia uma estação ativa, foi apresentado apenas os dados dessa. A Figura 23 demonstra a visualização deste requisito na ferramenta Kibana e a Figura 24 apresenta o mesmo requisito atendido na ferramenta Grafana.



Figura 23 – Mapa das estações no Kibana

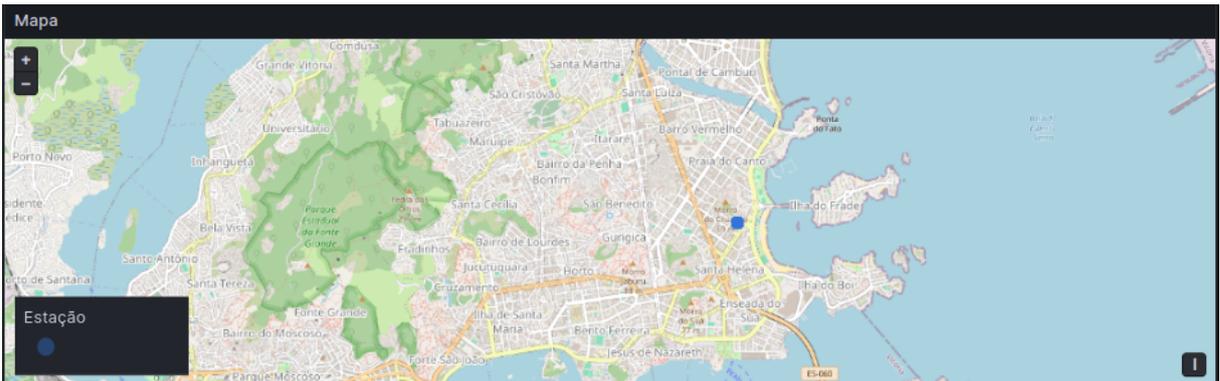


Figura 24 – Mapa das estações no Grafana

Desta forma, as ferramentas se mostraram equivalentes para a apresentação do mapa com as estações, já que foi possível configurar e plotar os dados de maneira muito similar tanto no Kibana, quanto no Grafana.

4.3.3 RF03: Apresentar as medições dos sensores das estações

Este requisito foi atendido tanto na ferramenta Grafana, quanto Kibana. Para atender esse requisito utilizamos o padrão Gauge de visualização, que permite a visualização de uma métrica de maneira simples. Como a especificação desse requisito visava a visualização online, basta definir o período de visualização como "último 5 minutos", que o ultimo valor armazenado no banco seria apresentado.

A Figura 25 demonstra a configuração para obter os dados online no Kibana, e a Figura 26 apresenta a mesma configuração desenvolvida na plataforma Grafana. Outro ponto a se atentar é o período de atualização para dados, que pode ser configurado para recarregar o *dashboard* de acordo com o tempo definido, como é demonstrado também nas Figuras 25 e 26.

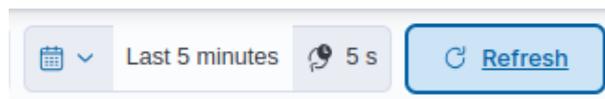


Figura 25 – Seleção de Tempo Kibana

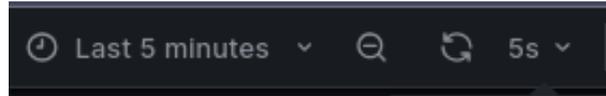


Figura 26 – Seleção de Tempo Grafana

Sendo assim, as ferramentas Grafana e Kibana atenderam a esse requisito de forma muito similar tanto na parametrização dos dados quanto na visualização.

4.3.4 RF04: Apresentar o histórico de medições dos sensores das estações

Para atender esse requisito utilizamos o padrão de visualização Histograma para plotar os dados no Grafana e o tipo Painel Line do Kibana. Essas tipos de painéis funcionam de maneira muito semelhante e permite a plotagem de um gráfico com dados históricos, conforme definidos nas Seções 4.1.2 e 4.2.2.

A plotagem dos dados acontece após a a seleção do período de tempo nos seletores de tempo de cada ferramenta de visualização. Uma vez que está selecionado o período de tempo necessário a ferramenta cruza os dados anteriormente configurados com o período de tempo selecionado. A Figura 27 apresenta a seleção de um período de tempo e a plotagem de um gráfico do tipo Histograma na plataforma Grafana. E a Figura 28 esta mesma configuração na plataforma Kibana.



Figura 27 – Histograma e Tempo Grafana



Figura 28 – Histograma e Tempo Kibana

Tanto o Grafana quanto o Kibana demonstraram abordagens similares na parametrização dos dados e na visualização, indicando uma equivalência na forma como lidam com esse requisito.

4.3.5 RF05: Apresentar o Índice de qualidade do Ar (IQAr) das estações

Para atender esse requisito foi necessário a criação de alguns painéis de visualização dos dados online, desta forma, utilizou-se o tipo de visualização Gauge em ambas as ferramentas Grafana e Kibana.

Tendo em vista que o índice de qualidade do ar (IQAr) se daria aos valores medidos pelos sensores de material particulado MP-10 e MP-25, foi criado painéis para a apresentação destes. A configuração conforme citada nas Seções de criação dos painéis, foi necessária para buscar os campos desejados. A Figura 29 demonstra a plotagem dos dados em suas representações na ferramenta Kibana, e Figura 30 apresenta está mesma plotagem representada na ferramenta Grafana.

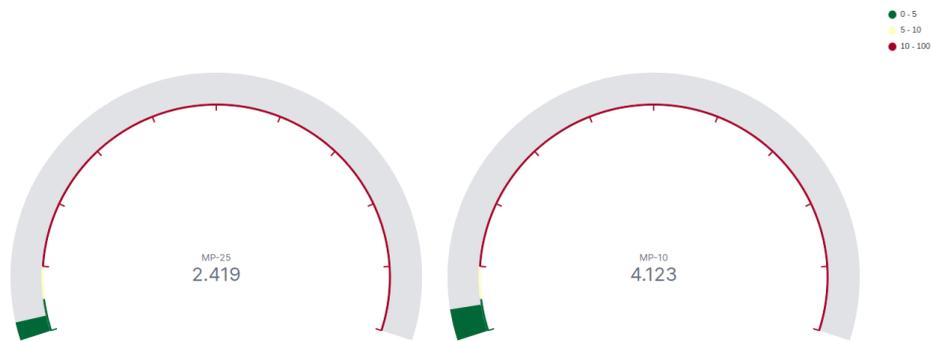


Figura 29 – IQAr online Kibana

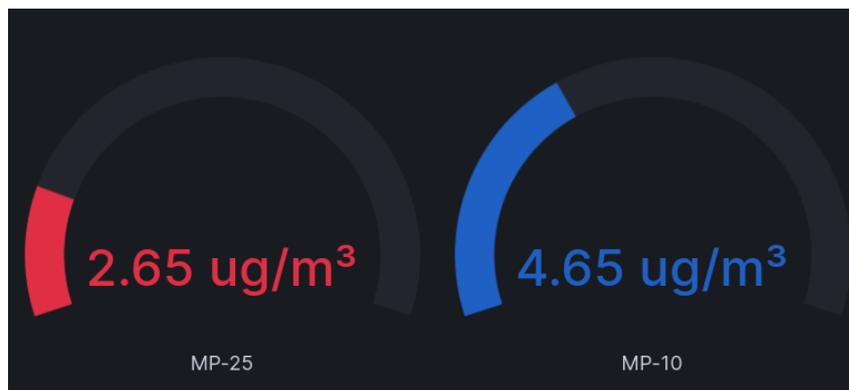


Figura 30 – IQAr online Grafana

A criação de painéis de visualização online utilizando o tipo de visualização Gauge, tanto no Grafana quanto no Kibana, foi eficaz para atender ao requisito de apresentação do Índice de Qualidade do Ar (IQAr) das estações. Os painéis permitiram uma representação clara dos valores medidos pelos sensores de material particulado MP-10 e MP-25. A configuração dos painéis, detalhada nas seções anteriores, garantiu a precisão dos dados exibidos. As figuras apresentadas ilustram a eficácia dessa abordagem em ambas as ferramentas, fornecendo uma maneira simples e direta de visualizar e analisar a qualidade do ar.

4.3.6 RF06: Apresentar o histórico do Índice de qualidade do Ar (IQAr) das estações

Para atender esse requisito utilizamos o padrão de visualização Histograma para plotar os dados no Grafana e o tipo Painel Line do Kibana. Essas tipos de painéis funcionam de maneira muito semelhante e permite a plotagem de um gráfico com dados históricos, conforme definidos nas Seções 4.1.2 e 4.2.2.

Tendo em vista que o índice de qualidade do ar(IQAr) se daria aos valores medidos pelos sensores de material particulado MP-10 e MP-25, foi criado painéis para a apresentação destes. A Figura 31 demonstra a plotagem dos dados em suas representações na ferramenta Kibana, e Figura 32 apresenta está mesma plotagem representada na ferramenta Grafana.

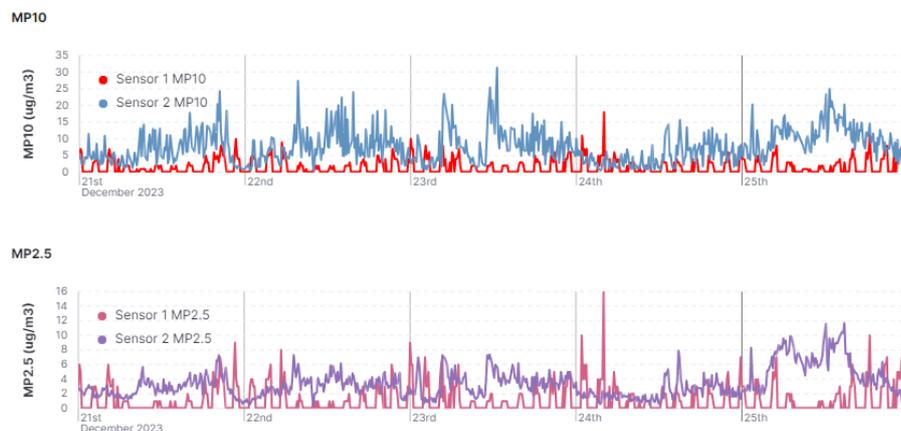


Figura 31 – IQAr Histórico Kibana

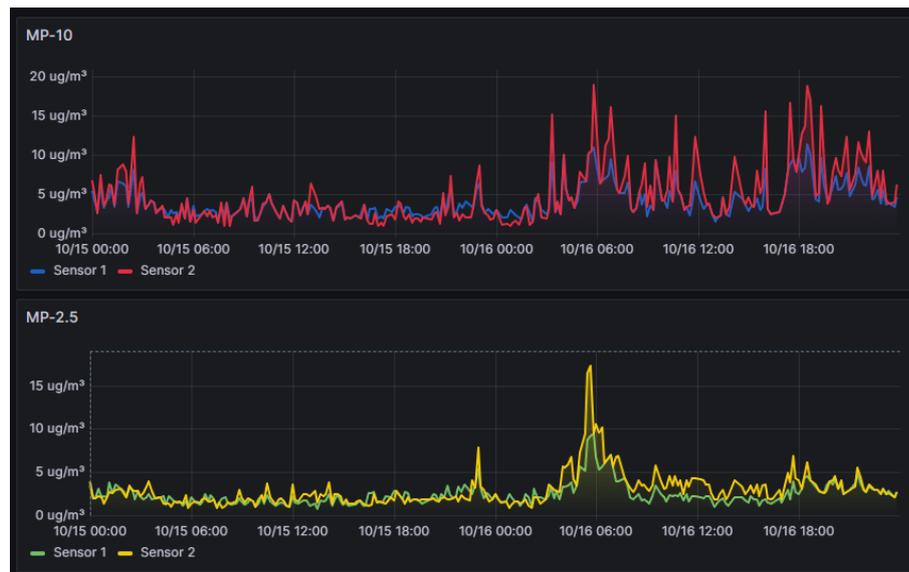


Figura 32 – IQAr Histórico Grafana

A parametrização dos dados e a visualização foram tratadas de maneira similar tanto no Grafana quanto no Kibana, indicando que ambas as ferramentas são equivalentes na forma como atendem a esse requisito.

4.3.7 RF07: Apresentar o Índice de calor das estações

Para cumprir esse requisito, foi necessário criar painéis de visualização dos dados online. Foram utilizados painéis do tipo Gauge nas ferramentas Grafana e Kibana para garantir uma exibição eficaz dos dados.

O Índice de Calor é calculado a partir dos valores medidos pelos sensores de temperatura e umidade. Dessa forma, foram criados painéis específicos para a apresentação desses dados. As configurações necessárias para buscar os campos desejados foram descritas nas Seções 4.1.2 e 4.2.2. A Figura 33 ilustra a plotagem dos dados na ferramenta Kibana, e a Figura 34 mostra a mesma plotagem na ferramenta Grafana.

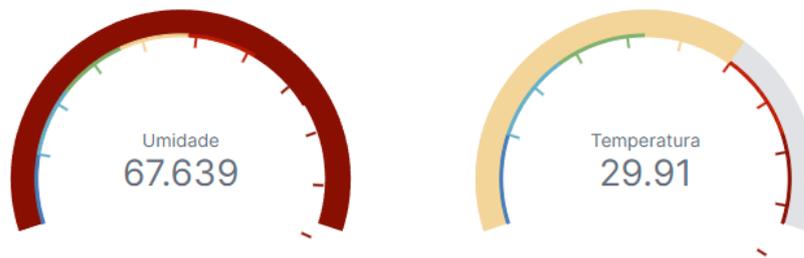


Figura 33 – Índice de Calor online - Kibana

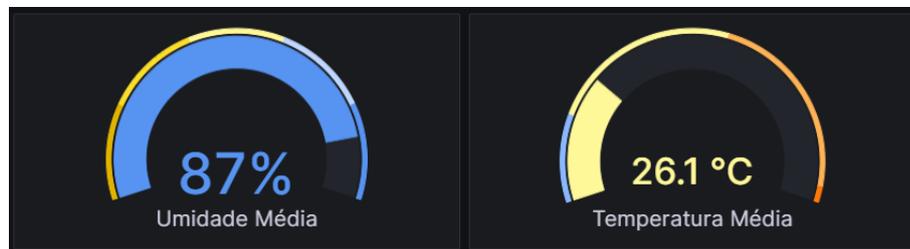


Figura 34 – Índice de Calor online - Grafana

Grafana e Kibana apresentaram métodos similares na parametrização e visualização dos dados, demonstrando uma equivalência na forma como lidam com este requisito.

4.3.8 RF08: Apresentar o histórico do Índice de calor das estações

Para atender a esse requisito, utilizamos o padrão de visualização Histograma para plotar os dados no Grafana e o tipo de painel Line no Kibana. Esses tipos de painéis funcionam de maneira bastante semelhante, permitindo a criação de gráficos com dados históricos, conforme detalhado nas Seções 4.1.2 e 4.2.2.

Considerando que o Índice de Calor é determinado pelos valores medidos pelos sensores de temperatura e umidade, foram desenvolvidos painéis específicos para a apresentação desses dados. A Figura 35 mostra a plotagem dos dados na ferramenta Kibana, enquanto a Figura 36 apresenta a mesma plotagem na ferramenta Grafana.

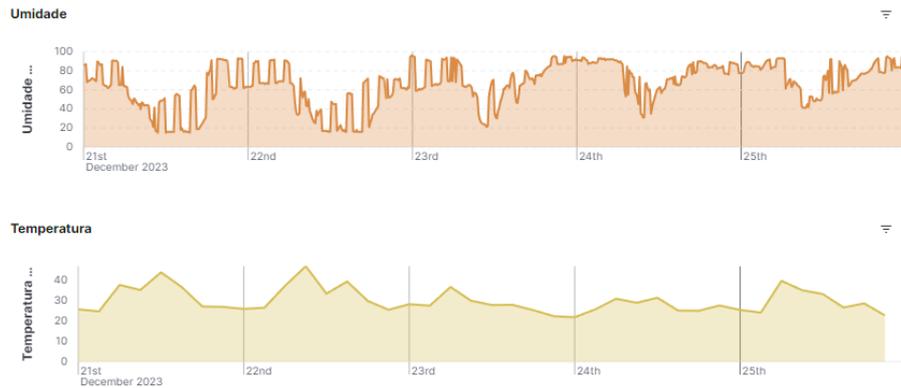


Figura 35 – Índice de Calor Histórico - Kibana

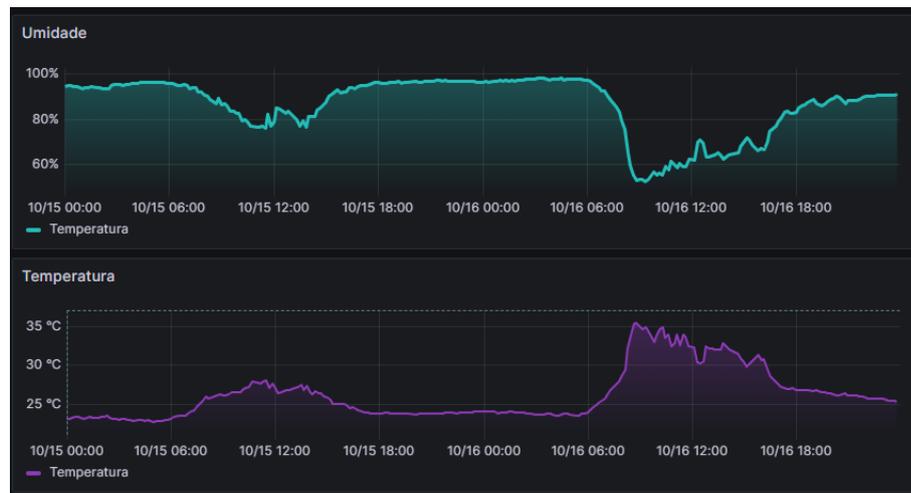


Figura 36 – Índice de Calor Histórico - Grafana

4.3.9 RF09: Apresentar a localização do usuário no mapa

Este requisito não foi atendido em nenhuma das plataformas, pois os dados necessários para a plotagem da localização do usuário não estão disponíveis no Elasticsearch utilizado como fonte de dados. A falta dessas informações impede a apresentação precisa da localização do usuário. No entanto, uma vez que os dados forem disponibilizados, a implementação dessa funcionalidade será direta e semelhante ao requisito 2 ("RF02: Apresentar um mapa da cidade com todas as estações").

4.3.10 RF10: Apresentar os melhores e piores índices e parâmetros das últimas 24 horas

Para atender esse requisito foi necessário criar visualizações que apresentassem os dados das últimas 24 horas, apresentando os valores máximos e mínimos de cada índice. Desta forma, foi utilizado as visualizações do tipo Gauge para a plotagem dos dados, separando-os pelos valores mínimos, máximos e também a média de todos os registros.

A Figura 37 mostra a plotagem dos dados na ferramenta Grafana, enquanto a Figura 38 apresenta a mesma plotagem na ferramenta Kibana.



Figura 37 – Max, Min e Média - Grafana



Figura 38 – Max, Min e Média - Grafana

4.3.11 RF11: Apresentar o histórico dos melhores e piores índices e parâmetros

Para atender esse requisito utilizou-se os mesmos painéis de visualização criados para atender o requisito RF10, alterando apenas o período de tempo para a plotagem dos dados.

Uma vez que se altera as datas de início e fim da busca, os dados serão plotados de acordo com esses valores fazendo com que seja possível apresentar os melhores e piores índices e parâmetros de todo o período cujo foi coletado dados.

A Figura 37 mostra a plotagem dos dados na ferramenta Grafana, enquanto a Figura 38 apresenta a mesma plotagem na ferramenta Kibana.

4.3.12 RF12: Apresentar o dia da semana com os melhores e piores índices e parâmetros

Para atender a esse requisito, utilizamos o padrão de visualização Histograma para plotar os dados no Grafana e o tipo de painel Line no Kibana. Esses tipos de painéis funcionam de maneira bastante semelhante, permitindo a criação de gráficos com dados históricos, conforme detalhado nas Seções 4.1.2 e 4.2.2.

Uma vez que os dados estão plotados nesse painéis, pode ser selecionado o período de tempo corresponde a semana que o usuário deseja obter os dados. Dessa forma, a identificação do dia da semana com os melhores e piores índices e parâmetros fica a cargo do usuário ao analisar os gráficos e encontrar essas informações.

A Figura 27 demonstra como selecionar o período de tempo em um histograma para conseguir fazer a análise dos dados na ferramenta Grafana, e a Figura 28 apresenta na ferramenta Kibana.

4.3.13 RF13: Apresentar o horário com os melhores e piores índices e parâmetros para cada estação

Para atender a esse requisito, utilizamos o padrão de visualização Histograma para plotar os dados no Grafana e o tipo de painel Line no Kibana. Esses tipos de painéis funcionam de maneira bastante semelhante, permitindo a criação de gráficos com dados históricos, conforme detalhado nas Seções 4.1.2 e 4.2.2.

Uma vez que os dados estão plotados nesse painéis, pode ser selecionado o período de tempo corresponde ao dia que o usuário deseja obter os dados. Dessa forma, a identificação do horário do dia com os melhores e piores índices e parâmetros fica a cargo do usuário ao analisar os gráficos e encontrar essas informações.

A Figura 27 demonstra como selecionar o período de tempo em um histograma para conseguir fazer a análise dos dados na ferramenta Grafana, e a Figura 28 apresenta na ferramenta Kibana.

A Tabela5 apresenta de maneira sintetizada os requisitos e se foram atendidos ou não nas ferramentas Grafana e Kibana.

Requisito	Grafana	Kibana
RF01: Armazenar as medições dos sensores	Atendido	Atendido
RF02: Apresentar um mapa da cidade com todas as estações	Atendido	Atendido
RF03: Apresentar as medições dos sensores	Atendido	Atendido
RF04: Apresentar o histórico de medições dos sensores	Atendido	Atendido
RF05: Apresentar o IQAr das estações	Atendido	Atendido
RF06: Apresentar o histórico do IQAr das estações	Atendido	Atendido
RF07: Apresentar o Índice de Calor das estações	Atendido	Atendido
RF08: Apresentar o histórico do Índice de Calor	Atendido	Atendido
RF09: Apresentar a localização do usuário no mapa	Não Atendido	Não Atendido
RF10: Apresentar os melhores e piores índices das últimas 24 horas	Atendido	Atendido
RF11: Apresentar o histórico dos melhores e piores índices	Atendido	Atendido
RF12: Apresentar o dia da semana com melhores e piores índices	Atendido	Atendido
RF13: Apresentar o horário com melhores e piores índices	Atendido	Atendido

Tabela 5 – Atendimento dos requisitos da aplicação 1

4.4 Comparação com Requisitos Definidos para Aplicação 2

Esta seção apresenta os resultados obtidos a partir dos requisitos definidos na Tabela 3. Para cada requisito foi feita uma análise comparativa nas duas ferramentas utilizando os painéis de visualização criados e os recursos disponíveis em cada uma das ferramentas Grafana e Kibana.

4.4.1 RF01: Apresentar um mapa da cidade com todos os veículos

Para atender esse requisito foi necessário a criação de um painel do tipo Mapa para a plotagem dos dados da aplicação. As Seções 4.1.1 e 4.2.1 descrevem como foi criado o painel do tipo mapa nas ferramentas Grafana e Kibana.

Desta forma, este requisito foi atendido apresentando os dados de cada caminhão de lixo separadamente plotando suas localizações a partir dos registros existentes no banco de dados.

A Figura 39 apresenta o mapa criado para atender esse requisito na ferramenta Kibana, enquanto a Figura 40 apresenta o mesmo mapa criado na ferramenta Grafana.



Figura 39 – Mapa dos Veículos - Kibana

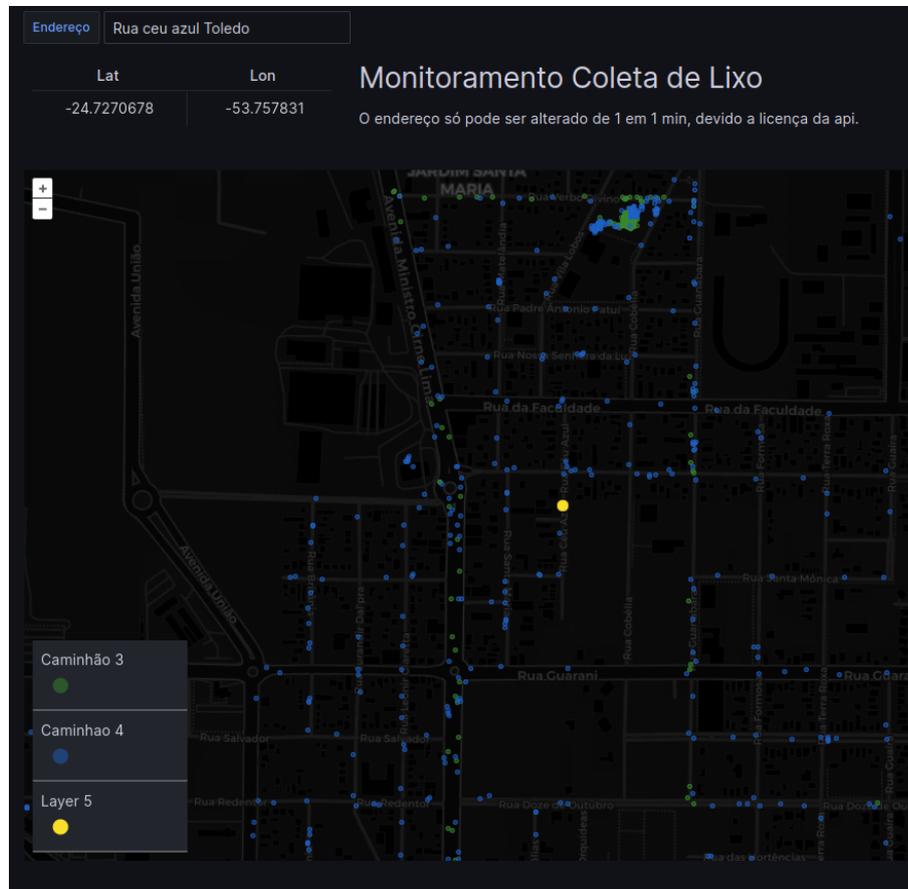


Figura 40 – Mapa dos Veículos - Kibana

4.4.2 RF02: Apresentar a localização do usuário no mapa

Para atender este requisito, inicialmente tentou-se obter a localização do usuário diretamente dos dispositivos; no entanto, devido a limitações técnicas e de privacidade, não foi possível implementar essa funcionalidade completamente.

Como alternativa, na ferramenta Grafana, foi adotada uma abordagem na qual o usuário pode fornecer seu endereço, e através de uma API externa o ponto correspondente ao endereço fornecido é plotado no mapa, representando a localização aproximada do endereço digitado. Essa integração está descrita na Seção 4.1.1.

É importante ressaltar que, devido à falta de suporte a APIs externas, não foi viável implementar essa funcionalidade na ferramenta Kibana.

Apesar de não ser uma solução ideal, essa abordagem na ferramenta Grafana permite ao usuário visualizar sua posição relativa no contexto do mapa da cidade, facilitando a interação com a aplicação.

4.4.3 RF03: Apresentar a localização de veículo específico no mapa

Para atender este requisito, foi utilizado o mesmo mapa implementado para os requisitos anteriores. O mapa exibe a localização de todos os veículos, proporcionando uma visão abrangente da frota.

Nas Figuras 41 e 42, são apresentadas as visualizações do mapa com a localização de um veículo específico, nas ferramentas Kibana e Grafana, respectivamente.

A interação com o mapa é facilitada através da funcionalidade em que o usuário pode clicar em cima do veículo desejado na legenda. Ao fazer isso, os outros veículos são ocultados, permitindo ao usuário visualizar exclusivamente a rota e a posição do veículo selecionado.

4.4.4 RF04: Apresentar o histórico de rotas dos veículos

Para atender este requisito, é utilizado o mesmo mapa implementado para os requisitos anteriores. Além de exibir a localização dos veículos, o sistema permite visualizar o histórico de rotas dos veículos em períodos anteriores.

Através da mesma interface de visualização do mapa, os usuários podem alterar o período de tempo dos dados obtidos. Isso possibilita a apresentação do histórico de rotas de um certo período selecionado nos campos de tempo disponíveis na interface.

Essa funcionalidade permite aos usuários analisar o comportamento e os padrões de movimentação dos veículos ao longo do tempo. Por exemplo, é possível identificar rotas frequentes, horários de maior atividade, ou até mesmo desvios não planejados.

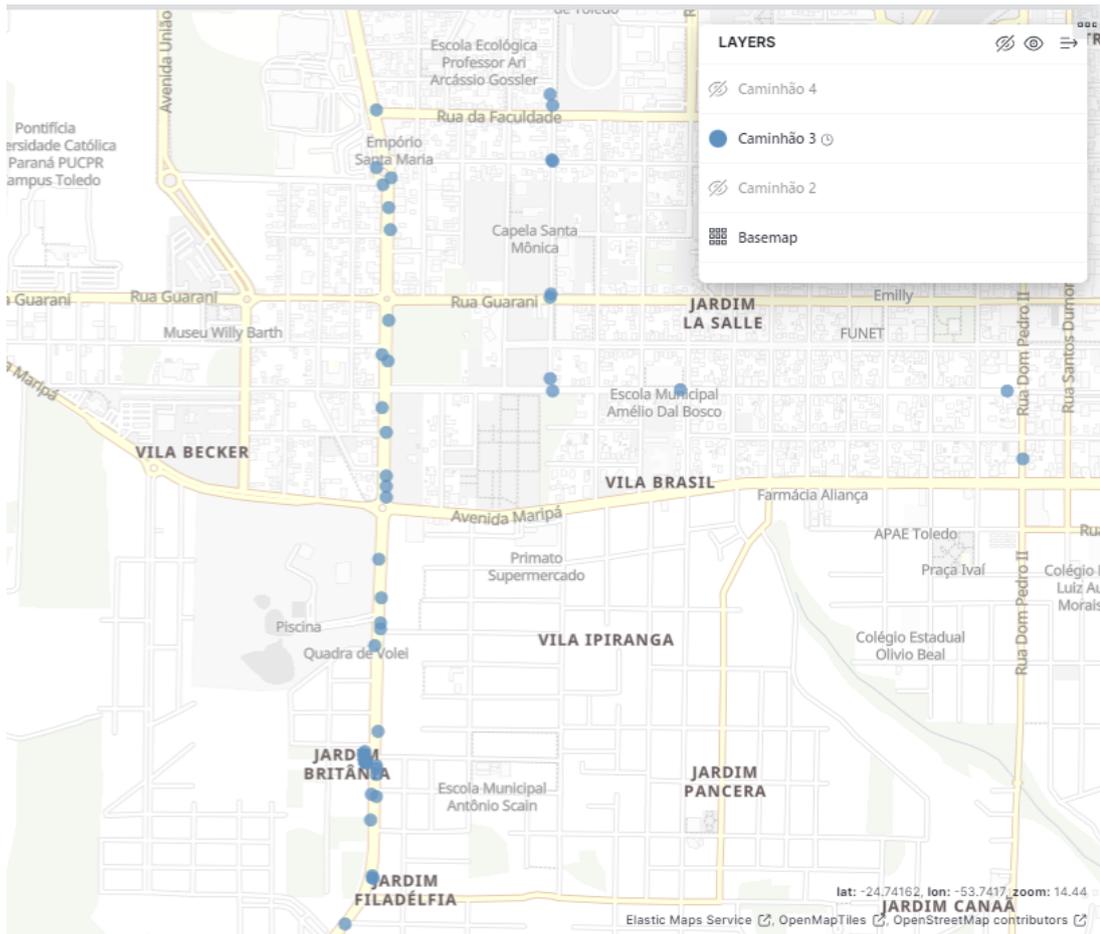


Figura 41 – Veículo Específico - Kibana

4.4.5 RF05: Apresentar o histórico de coleta/passagem de veículo em determinada localização

Para atender este requisito, é utilizado o mesmo mapa implementado para os requisitos anteriores. O sistema permite aos usuários visualizar o histórico de coleta ou passagem de veículos em determinadas localizações ou regiões.

Na interface do mapa, o usuário tem a liberdade de navegar e procurar a localização específica ou região de interesse. Utilizando as ferramentas de zoom e pan disponíveis, é possível explorar o mapa e verificar se um caminhão de lixo passou ou não por determinada área.

Além disso, o usuário tem a capacidade de escolher o período de tempo desejado para analisar o histórico de coleta ou passagem de veículos. Através dos filtros de tempo disponíveis na interface, é possível especificar um dia/horário específico para verificar se o caminhão passou naquele lugar em um determinado momento.

Na ferramenta Grafana, há um auxílio adicional fornecido por uma API externa. O usuário pode digitar um endereço na interface para se localizar melhor no mapa, facilitando a busca por áreas específicas de interesse.

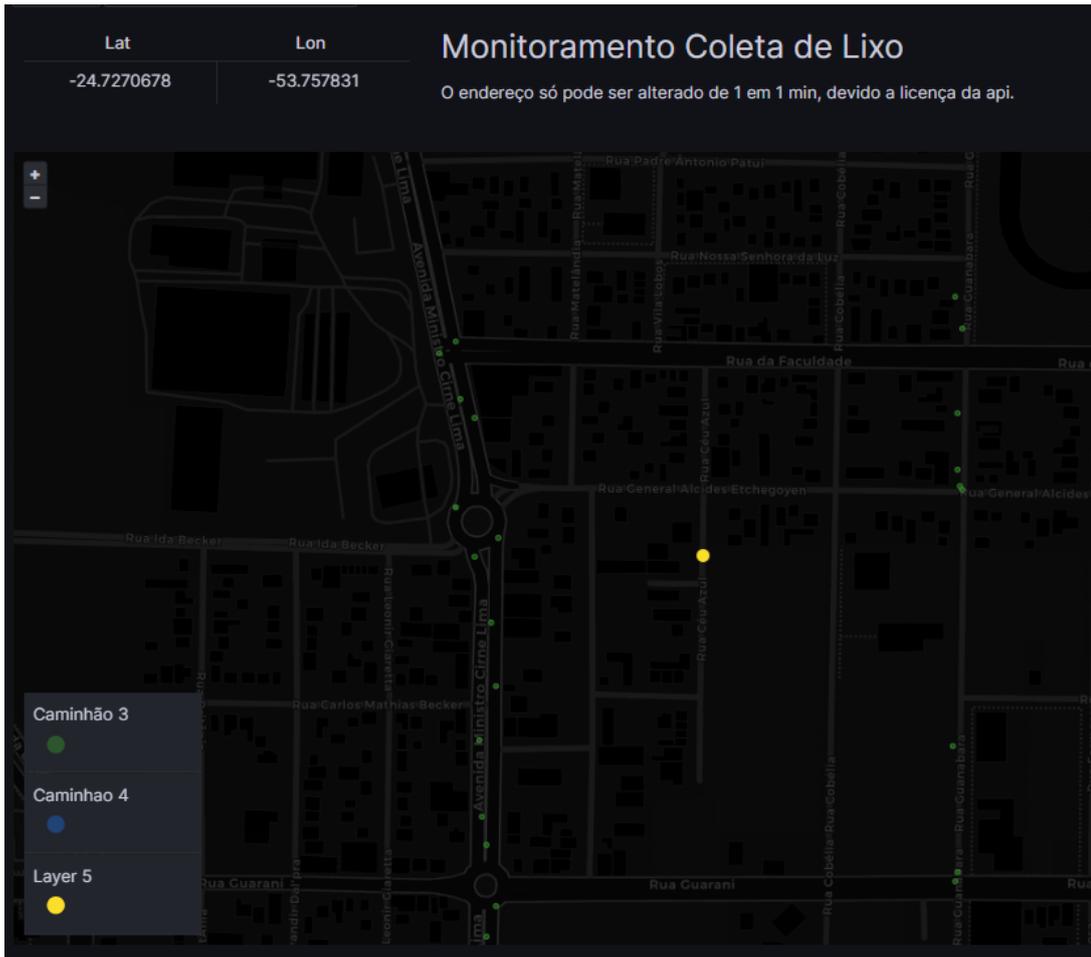


Figura 42 – Veículo Específico - Grafana

No entanto, é importante ressaltar que na ferramenta Kibana não há suporte para essa funcionalidade adicional da API externa. Portanto, o usuário deve confiar nas funcionalidades padrão do mapa para navegar e explorar as áreas de interesse.

A Tabela 6 apresenta de maneira sintetizada os requisitos e se foram atendidos ou não nas ferramentas Grafana e Kibana.

Requisito	Grafana	Kibana
RF01: Apresentar um mapa da cidade com todos os veículos	Atendido	Atendido
RF02: Apresentar a localização do usuário no mapa	Atendido (via API externa)	Não Atendido
RF03: Apresentar a localização de veículo específico no mapa	Atendido	Atendido
RF04: Apresentar o histórico de rotas dos veículos	Atendido	Atendido
RF05: Apresentar o histórico de veículo em determinada localização	Atendido (via API externa)	Atendido (sem API externa)

Tabela 6 – Atendimento dos Requisitos da aplicação 2

Embora a maioria dos requisitos tenham sido apresentados de maneira clara através de painéis e *dashboards*, a parametrização para visualização de dados em um certo período de tempo ainda é um ponto complexo nas ferramentas Grafana e Kibana. O ajuste dos períodos de tempos é confuso nas ferramentas, pois foram feitas pra serem utilizadas por profissionais da área de tecnologia. Para um usuário final sem treinamento e familiaridade com as ferramentas, a chance de frustração é grande por não conseguir parametrizar o período de tempo de maneira adequada. Com treinamento e com o uso recorrente das ferramentas o usuário deve ser capaz de conseguir alterar os períodos de tempo sem dificuldades.

4.5 Avaliação das Documentações das Ferramentas

Para Grafana, a documentação oficial é clara e abrangente, cobrindo desde a instalação até o uso avançado. Além disso, existem muitos tutoriais e guias práticos disponíveis em blogs e sites especializados. A comunidade de Grafana é ativa, oferecendo suporte através de fóruns e canais de discussão, o que facilita a resolução de problemas e a troca de informações. Além disso a ferramenta Grafana é *open-source* o que faz com que muitos usuário desenvolvedores, implementem *plugins* e extensões. No desenvolvimento trabalho, todos os tópicos em que se fez necessário buscar ajuda na internet, sempre encontrou-se soluções no forum oficial da comunidade do Grafana¹.

No caso do Kibana, a qualidade de documentação oficial é abrangente porém não clara o suficiente para que alguém com pouca experiência consiga utilizar os recursos necessários. Em contra-partida há diversos tutoriais e recursos educacionais disponíveis online através da comunidade. A Kibana é uma ferramenta um pouco mais complexa de se utilizar, e com a qualidade das documentações não sendo boa o suficiente a comunidade busca se ajudar utilizando o fórum da própria Elastic².

4.6 Externalização de Dashboards

Apesar de ser possível acessar diretamente as plataformas Grafana e Kibana para a visualização dos *dashboards* criados, as ferramentas disponibilizam recursos para publicar ou incorporar os painéis em uma página WEB. A ferramenta Grafana disponibiliza duas formas de publicação de um *dashboard*: uma é a criação de um link externo para que qualquer usuário consiga acessar diretamente a página do *dashboard*; e a outra é permitir a incorporação, onde o usuário poderá utilizar o link público em um arquivo HTML com a tag *iframe*. Para utilizar a visualização incorporada em outra página WEB, é necessário ativar a configuração '*allow_embedding*', caso contrário a ferramenta Grafana bloqueia a visualização externa. A Figura 43 demonstra a incorporação de um *dashboard* em um arquivo HTML com a tag *iframe*.

¹ <https://community.grafana.com/>

² <https://discuss.elastic.co/>

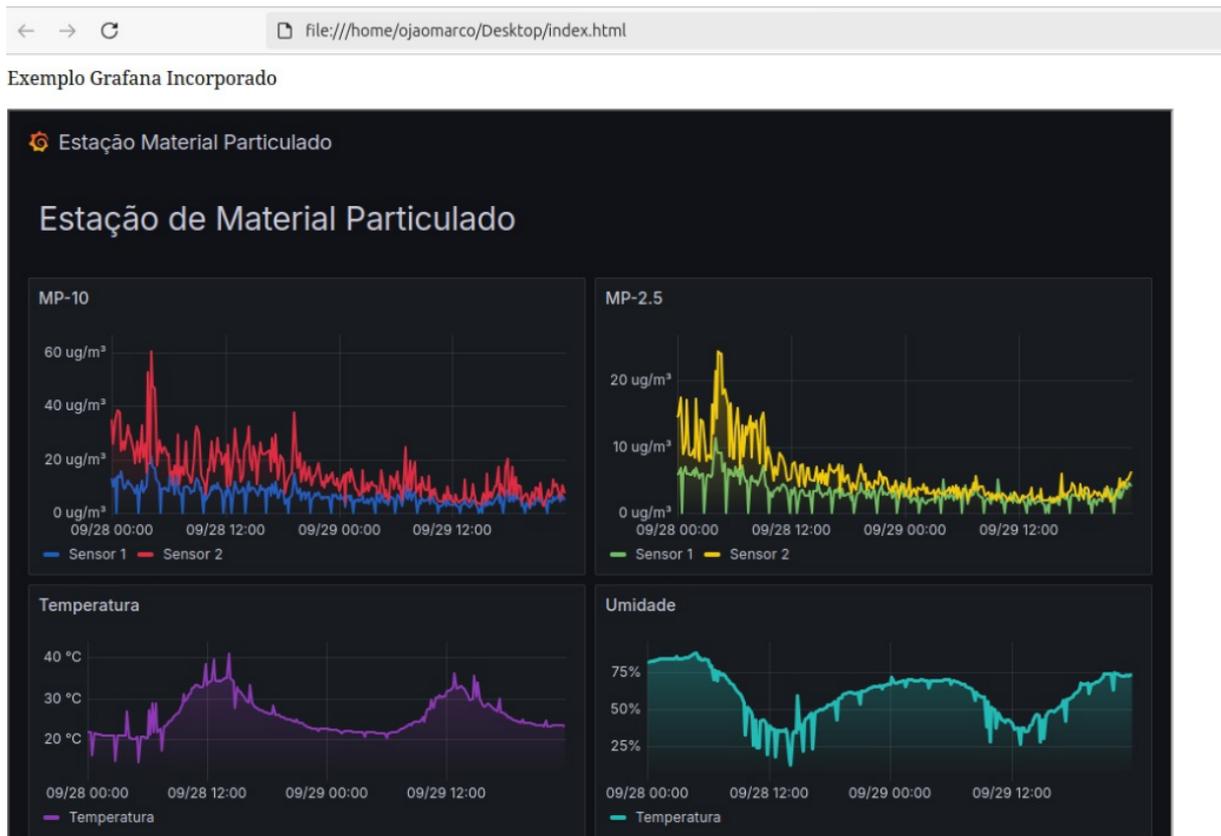


Figura 43 – Exemplo Grafana Incorporado

A ferramenta Kibana apresenta os mesmos recursos que a Grafana, a possibilidade de gerar um link público, e a possibilidade de incorporar um *dashboard* em um arquivo HTML com a tag *iframe*. A Kibana não requer configurações para conseguir fazer a exportação de um *dashboard*, porém pede a autenticação do usuário para a visualização dos painéis. A Figura 44 demonstra a incorporação de um *dashboard* Kibana em um arquivo HTML com a tag *iframe*.

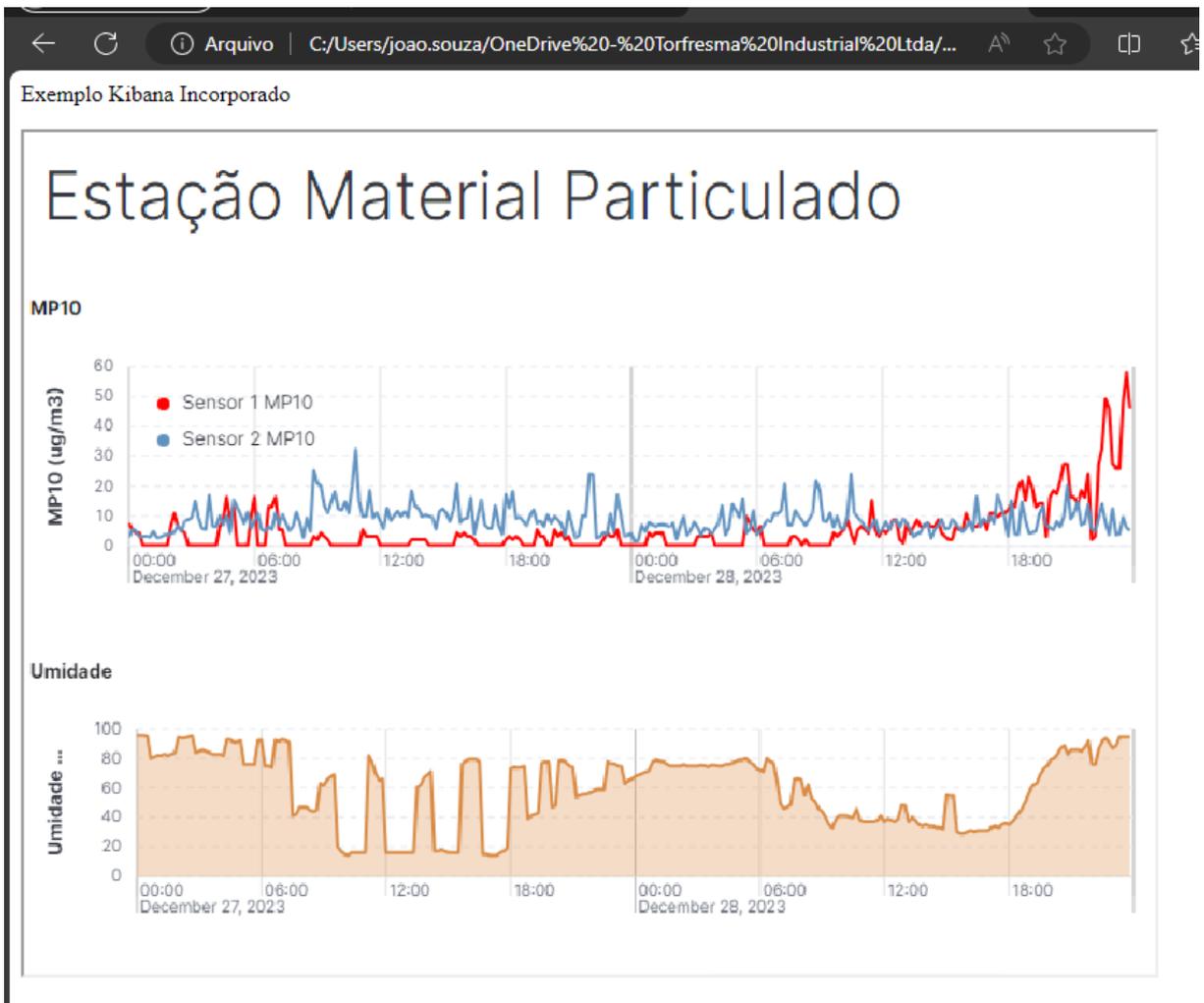


Figura 44 – Exemplo Kibana Incorporado

5 CONCLUSÃO

Este estudo visou a comparação das ferramentas de visualização de dados Grafana e Kibana para aplicações de cidades inteligentes. Uma das aplicações aborda o rastreamento dos caminhões de coleta seletiva (ROSSATO; SPANHOL; CAMARGO, 2020), e a outra está voltada para o monitoramento da qualidade do ar (PASTÓRIO *et al.*, 2022). Essas aplicações serviram com fonte de dados para a realização deste estudo comparativo.

Para a comparação foram utilizados parâmetros como a qualidade da documentação, a facilidade de configuração e uso das ferramentas, além dos requisitos específicos definidos em um trabalho de mestrado em desenvolvimento que visa avaliar plataformas para cidades inteligentes (PAGANINI, 2024). Resultados obtidos se mostraram satisfatórios, sendo possível visualizar e analisar os dados seguindo os requisitos de cada aplicação.

As ferramentas Grafana e Kibana permitiram a visualização dos dados de diferentes formas, transformando-os em informações úteis. Como foi utilizado uma base de dados Elastic-Search a ferramenta Kibana leva certa vantagem quando o quesito é configuração e conexão por ser uma ferramenta nativa do arcabouço ELK. Já a ferramenta Grafana leva vantagem em outros pontos, como a personalização de painéis mais completa, a facilidade de se usar a ferramenta por ser muito intuitiva.

Apesar de terem vantagens, ambas ferramentas compartilham uma desvantagem que é a forma de configuração e parametrização dos períodos de tempo. Esse recurso funciona de uma maneira complexa e difícil de se entender tanto na ferramenta Kibana quanto na Grafana. Este é um ponto que poderia ser mais simples e voltado para o usuário final.

Ao fim deste trabalho chegou-se a conclusão que as duas ferramentas atendem o uso para visualização de dados de aplicações de cidades inteligentes, porém para dois públicos diferentes. A Kibana entrega um ambiente mais focado na análise dos dados por estar em conjunto direto com a base de dados, servindo mais para os desenvolvedores da aplicação e profissionais da área, sem que se trabalhe com o usuário final em contato com a ferramenta. Já a Grafana apresenta um ambiente mais comercial, podendo ser a interface direta do usuário com a aplicação já que é simples e completa.

5.1 Trabalhos Futuros

Este trabalho obteve resultados promissores e que podem ser estudados com mais profundidade em trabalhos futuros. Um caminho que pode ser seguido é avaliar as ferramentas visando principalmente o consumo de recursos computacionais, como tempo de requisição e a capacidade de lidar com grandes volumes de dados.

REFERÊNCIAS

- BAJER, M. Building an iot data hub with elasticsearch, logstash and kibana. p. 63–68, 2017.
- BARBOZA, R. M. Monitoramento voltado à cibersegurança em sistemas industriais. *In: BARBOZA, Rafael Menezes. Monitoramento voltado à cibersegurança em sistemas industriais. 2020. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná, Campo Mourão, 2020.* Campo Mourao, PR, Brasil: Universidade Tecnológica Federal do Paraná, 2020. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/26105>.
- BARRO, P. A. *et al.* A smart cities lorawan network based on autonomous base stations (bs) for some countries with limited internet access. **Future Internet**, MDPI, v. 11, n. 4, p. 93, 2019.
- CAMARGO, E. T. de; SPANHOL, F. A.; SOUZA, Á. R. Castro e. Deployment of a lorawan network and evaluation of tracking devices in the context of smart cities. **Journal of Internet Services and Applications**, SpringerOpen, v. 12, n. 1, p. 1–24, 2021.
- CHAKRABORTY, M.; KUNDAN, A. P. Grafana. *In: Monitoring Cloud-Native Applications: Lead Agile Operations Confidently Using Open Source Software.* [S.l.]: Springer, 2021. p. 187–240.
- CHAN, N. A resource utilization analytics platform using grafana and telegraf for the savio supercluster. *In: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning).* New York, NY, USA: Association for Computing Machinery, 2019. (PEARC '19). ISBN 9781450372275. Disponível em: <https://doi.org/10.1145/3332186.3333053>.
- ELASTIC. **ELK Customers.** 2021. Disponível em: <https://www.elastic.co/customers/>.
- ELASTIC. **ELK Elasticsearch.** 2021. Disponível em: <https://www.elastic.co/elasticsearch/>.
- ELASTIC. **ELK Kibana.** 2021. Disponível em: <https://www.elastic.co/kibana/>.
- ELASTIC. **ELK Logstash.** 2021. Disponível em: <https://www.elastic.co/logstash/>.
- GRAFANA. **Grafana Site.** 2023. <https://grafana.com/grafana/>. Acesso em: 27/05/2023.
- Grafana Document. **Plugin Management.** 2021. <https://grafana.com/docs/grafana/latest/administration/plugin-management/>. Acesso em: 27/05/2023.
- KHANH, D. B. **Data Visualization with Integration of Grafana and Quanta Platform.** 2021. 50 p.
- KIRIMTAT, A. *et al.* Future trends and current state of smart city concepts: A survey. **IEEE Access**, v. 8, p. 86448–86467, 2020.
- LAKHARA, S.; MISHRA, N. Desktop full-text searching based on lucene: A review. *In: 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI).* [S.l.: s.n.], 2017. p. 2434–2438.
- PAGANINI, D. Scalability evaluation of the sentilo platform running on containerized resources. Artigo em publicação. 2024.

PASTÓRIO, A. F. *et al.* A machine learning-based approach to calibrate low-cost particulate matter sensors. *In: 2022 XII Brazilian Symposium on Computing Systems Engineering (SBESC)*. [S.l.: s.n.], 2022. p. 1–8.

PRIAMBODO, R.; KADARINA, T. M. Monitoring self-isolation patient of covid-19 with internet of things. p. 87–91, 2020.

ROSSATO, J.; SPANHOL, F.; CAMARGO, E. Implantação e avaliação de uma rede sem-fio de longo alcance e baixa potência para cidades inteligentes. *In: Anais do IV Workshop de Computação Urbana*. Porto Alegre, RS, Brasil: SBC, 2020. p. 192–205. ISSN 2595-2706. Disponível em: <https://sol.sbc.org.br/index.php/courb/article/view/12363>.

SANTANA, E. F. Z. *et al.* Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 50, n. 6, nov 2017. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3124391>.

SARTURI, G. G. Monitoramento, controle e aquisição de dados de uma estufa utilizando esp32, mysql e grafana. *In: SARTURI, G.G. Monitoramento, controle e aquisição de dados de uma estufa utilizando ESP32, MySQL e GRAFANA. 2022. 52 p. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) - Universidade Federal de Santa Maria, Santa Maria, RS, 2022*. Santa Maria, RS, Brasil: Universidade Federal de Santa Maria, 2022. Disponível em: <http://repositorio.ufsm.br/handle/1/26098>.

SINAEPOURFARD, A. *et al.* Estimating smart city sensors data generation. *In: 2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. [S.l.: s.n.], 2016. p. 1–8.

SÁNCHEZ-CORCUERA, R. *et al.* Smart cities survey: Technologies, application domains and challenges for the cities of the future. **International Journal of Distributed Sensor Networks**, v. 15, n. 6, p. 1550147719853984, 2019. Disponível em: <https://doi.org/10.1177/1550147719853984>.

THU, M. Y. *et al.* Smart air quality monitoring system with lorawan. p. 10–15, 2018.

Universidade Tecnológica Federal do Paraná. **Acordo de Cooperação Técnica Nº 001/2020**. 2020. Online: SEI UTFPR. Disponível no processo SEI 23064.000768/2020-96. Disponível em: https://sei.utfpr.edu.br/sei/controlador.php?acao=documento_conferir&id_protocolo=23064.000768%2F2020-96.