



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**LUCAS CHAVES SALUSTIANO**

**UM ESTUDO COMPARATIVO ENTRE FERRAMENTAS DE  
MONITORAMENTO DE CONTÊINERES DOCKER**

**CAMPINA GRANDE - PB**

**2023**

**LUCAS CHAVES SALUSTIANO**

**UM ESTUDO COMPARATIVO ENTRE FERRAMENTAS DE  
MONITORAMENTO DE CONTÊINERES DOCKER**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**Orientador: Professor Dr. Reinaldo Cezar de Moraes Gomes.**

**CAMPINA GRANDE - PB**

**2023**

**LUCAS CHAVES SALUSTIANO**

**UM ESTUDO COMPARATIVO ENTRE FERRAMENTAS DE  
MONITORAMENTO DE CONTÊINERES DOCKER**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Reinaldo Cezar de Moraes Gomes  
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Robert Kalley Cavalcanti de Menezes  
Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni  
Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 14 de Fevereiro de 2023.**

**CAMPINA GRANDE - PB**

## **ABSTRACT**

The use of containers has been widely adopted in the technology industry due to their flexibility and scalability. However, container monitoring is a crucial task to ensure system availability and performance. Therefore, there is a wide variety of monitoring tools available for various use cases and scopes. In this study, two of these tools, Prometheus and NetData, will be compared, both being pointed out by the Cloud Native Computing Foundation (CNCF) as two of the projects with the most contributors in 2022 [4].

The evaluation criteria will be five: real-time monitoring, baseline performance indicator, network performance monitoring, data visualization, and alerting. In the end, it was concluded that NetData has a slight advantage over Prometheus because it is more limited in aspects such as network performance monitoring and data visualization.

# Um Estudo Comparativo Entre Ferramentas de Monitoramento Contêineres Docker

Lucas Chaves Salustiano  
lucas.salustiano@ccc.ufcg.edu.br

Reinaldo Cezar de Moraes Gomes  
reinaldo@computacao.ufcg.edu.br

## RESUMO

A utilização de contêineres tem sido amplamente adotada na indústria de tecnologia devido à sua flexibilidade e escalabilidade. No entanto, o monitoramento dos contêineres é uma tarefa crucial para garantir a disponibilidade e o desempenho do sistema. Por isso, há uma ampla variedade de ferramentas de monitoramento disponíveis, para os mais variados casos de uso e escopos. Neste estudo, serão comparadas duas dessas ferramentas, o Prometheus e o NetData, ambas sendo apontadas pela Cloud Native Computing Foundation (CNCF) como sendo dois dos projetos com mais contribuidores de 2022 [4].

Os critérios de avaliação serão cinco: Monitoramento em tempo real, indicador base de performance, monitoramento de performance de rede, visualização de dados e alerta. Ao final, concluiu-se que o NetData obtém leve vantagem sobre o Prometheus por este ser mais limitado em aspectos como monitoramento de performance de rede e visualização de dados.

## Palavras-chave

Contêineres, monitoramento, métricas, recursos.

Repositório com os artefatos utilizados nos experimentos:  
<https://github.com/lucasalustiano/TCC>

## 1. INTRODUÇÃO

O monitoramento de contêineres é o processo de coleta contínua de métricas e observação da integridade de aplicativos em contêineres e ambientes de microsserviços, a fim de manter a integridade, melhorar o desempenho e garantir que estejam operando sem problemas. Os contêineres se tornaram uma das formas mais populares na implantação de aplicativos trazendo vários pontos positivos, entre eles, o aumento da portabilidade das aplicações desenvolvidas e flexibilidade operacional.

Segundo o survey realizado pela Cloud Native Computing Foundation (CNCF) em 2021 [1], 93% dos participantes afirmaram que estão utilizando, ou estão planejando utilizar, contêineres em seus respectivos ambientes de produção.

Com tamanha utilização de contêineres, torna-se necessário atentar-se para aspectos como monitoramento e observabilidade dos contêineres visto que são fundamentais para manter a integridade do sistema. Todavia nesse contexto, as ferramentas de monitoramento tradicionais não são adequadas dada a natureza efêmera dos contêineres visto que elas monitoram os recursos e a infraestrutura no nível do host. Por outro lado, as soluções de monitoramento baseadas em contêiner fornecem recursos robustos para rastrear possíveis falhas, bem como insights granulares sobre

o comportamento do contêiner, que são altamente dinâmicos e exigem um monitoramento mais rigoroso, a fim de manter a integridade da aplicação e dos recursos.

## 2. REVISÃO DE LITERATURA

Durante a revisão de literatura foi encontrada um comparativo entre ferramentas de monitoramento de contêineres Docker feita por Silva [8]. Na ocasião, foi desenvolvida uma solução para resolver um problema enfrentado pela autora no sistema em que trabalhava, onde havia a necessidade de monitorar, mitigar e alertar os administradores de sistema da equipe sobre eventuais falhas relacionadas aos contêineres em que eram executados serviços críticos, essenciais para a equipe de desenvolvimento. A solução foi arquitetada para ser utilizada em uma stack swarm, uma ferramenta de administração de clusters para contêineres do próprio Docker. Todavia, o ponderamento entre as opções de ferramentas a serem utilizadas no monitoramento dos contêineres foi simples. Este trabalho foi idealizado com o objetivo de realizar um estudo comparativo qualitativo mais aprofundado. Há também outras abordagens possíveis na questão do monitoramento de contêineres, a exemplo da proposta de Pfeifer et al [9] que se utiliza de agentes inteligentes e técnicas de aprendizagem de máquina para decidir uma frequência eficiente de avaliação/monitoramento dos contêineres presentes no ambiente de forma autônoma e dinâmica. Porém esse trabalho não irá considerar essa abordagem.

## 3. BACKGROUND

Nas seguintes subseções serão apresentados alguns conceitos e aspectos motivadores sobre o monitoramento de contêineres bem como os problemas associados.

### 3.1 Conceitos

Nessa subseção serão apresentados conceitos necessários para familiarização com aspectos que serão abordados durante o restante do trabalho.

#### 3.1.1 Docker

O Docker é uma ferramenta que ajuda a desenvolver, enviar e executar aplicativos em contêineres, que são pequenos pacotes de software que contêm tudo o que é necessário para executar um aplicativo, incluindo código, bibliotecas e configurações. Isso permite que os aplicativos sejam facilmente movidos de um ambiente de desenvolvimento para outro, sem a necessidade de modificações. Em resumo, o Docker é uma maneira conveniente de empacotar e distribuir aplicativos, tornando o processo mais fácil, mais rápido e mais consistente.

### 3.1.2 Contêiner

Um contêiner é um software controlado que isola um ou mais processos, e todas as suas dependências, e o(s) executa(m) de forma isolada, rápida e consistente em vários ambientes de computação. Além disso, os contêineres compartilham o kernel e os recursos físicos com a máquina host, o que proporciona melhor desempenho por conta do gerenciamento único dos recursos. Outro ponto importante é que os contêineres são isolados uns dos outros, o que significa que um aplicativo em execução em um contêiner não pode afetar outro aplicativo em execução em outro contêiner.



Figura 1: Diagrama de contêineres. Através do Docker, a aplicação 1 e a aplicação 2 podem compartilhar os recursos de hardware da máquina host e ainda assim funcionarem distintamente sem impactarem uma na outra.

### 3.1.3 Hypervisor

Hypervisor é um software que cria e gerencia máquinas virtuais (VMs) em um ambiente de computação. Ele é responsável por compartilhar os recursos físicos do host entre as máquinas virtuais, isolando as VMs uns dos outros e fornecendo às máquinas virtuais para instalação de sistemas operacionais e aplicativos. Os hypervisors também fornecem recursos como gerenciamento de memória, CPU, disco e rede para as máquinas virtuais. Em geral, o hypervisor é a base para a virtualização, permitindo a criação e execução de múltiplas máquinas virtuais em um único host.

### 3.1.4 Máquina Virtual

O termo "máquina virtual" (VM) refere-se a um ambiente computacional virtualizado que possui sua própria CPU, memória, interface de rede e armazenamento obtidos por meio de virtualização de recursos físicos de uma máquina base, denominada host. Os recursos usados pela máquina virtual, também chamada de guest, são "separados" do hardware por uma ferramenta de software conhecida como hypervisor, que os fornece de forma adequada. Assim, a máquina host e a máquina guest dividem os recursos de hardware, de maneira isolada, de acordo com a especificação de recursos definido pelo usuário por meio do hypervisor.

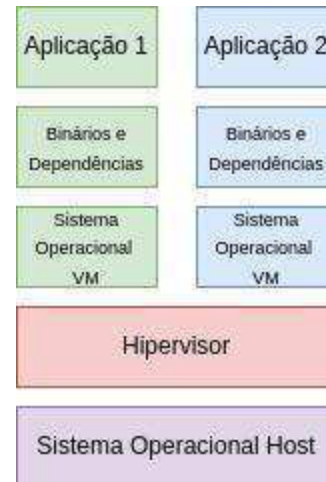


Figura 2: Diagrama de máquinas virtuais. Através de um dos vários hipervisors existentes, é possível criar múltiplas máquinas virtuais em um mesmo host. Cada máquina virtual agregará para si recursos de hardware da máquina host, mas funcionarão sem impactar uma na outra.

### 3.1.5 Monitoramento

O monitoramento de um sistema de computação é um processo de acompanhamento de performance e execução de aplicação por meio da coleta e análise de métricas do sistema. Essas métricas podem ir desde a porcentagem de carga da CPU até a quantidade de pacotes transmitidos/enviados em um determinado período de tempo. O monitoramento dessas métricas permite o administrador de sistemas (ou DevOps, SRE, etc) verificar o estado geral do sistema e tomar decisões, sejam elas, preventivas ou paliativas.

## 3.2 A Importância e a Necessidade de se Monitorar Contêineres

Para garantir uma boa gestão das aplicações em contêiner, é importante ter uma plataforma de monitoramento adequada que possa mostrar todos os dados de diferentes fontes em um único painel. Isso é necessário para juntar informações de diferentes planos e garantir que os dados possam ser facilmente correlacionados. A equipe responsável precisa planejar e pensar sobre como esse monitoramento será feito. Os dados de métricas são importantes para o gerenciamento dos recursos, como o escalonamento das operações. Eles são coletados em diferentes níveis, como no nível dos contêineres e na infraestrutura. Além disso, para solucionar problemas no aplicativo, é preciso ter informações específicas sobre o desempenho do aplicativo e detalhes de rastreamento.

A medida que um sistema de aplicativos cresce, é cada vez mais exigente monitorar tudo. Antigamente, os desenvolvedores criavam um serviço simples para realizar uma tarefa, e outros serviços trabalhavam juntos para realizar operações maiores. Hoje, cria-se serviços que incluem recursos de rede além da lógica de negócios, como comunicação entre serviços, gerenciamento de tráfego, repetições e afins. Mas, quando centenas de serviços são criados por várias equipes usando diferentes linguagens de programação, eles podem ficar incrivelmente complexos e difíceis de monitorar.

Para manter um ambiente tranquilo e maximizar o uso de recursos, a visibilidade e o monitoramento são cruciais. Ainda mais em sistemas containerizados, que são muito dinâmicos e precisam de monitoramento constante para garantir a integridade do sistema. Os problemas podem se propagar rapidamente por contêineres, aplicativos e toda a arquitetura, pois cada imagem de contêiner pode suportar várias instâncias ativas e porque novas imagens e versões são criadas rapidamente. Isso torna essencial identificar a origem de um problema assim que ele se manifesta.

A falta de um mecanismo que permita observar as informações sobre o estado de funcionamento do sistema, a qualidade dos serviços sendo oferecidos e a quantidade de recursos sendo consumidos podem acarretar em, por exemplo:

- Dificuldades operacionais – É um desafio extremamente complexo para desenvolvedores e administradores da infraestrutura compreender o funcionamento e operação de um sistema complexo. A existência de um problema, por menor que seja, pode ser de difícil resolução e demandar muito esforço e pessoas para resolver. Ferramentas de monitoramento são de grande valia nesses casos porque auxiliam em um diagnóstico consideravelmente mais rápido sobre os problemas em questão.
- Desafios com escalabilidade: Para sistemas containerizados, a capacidade de dimensionar rapidamente instâncias de aplicativos ou microsserviços, conforme necessário, é crucial. No entanto, o único método para determinar a demanda e a satisfação do cliente é por meio da observação das informações do sistema. Aumentar ou diminuir a escala tarde demais pode levar a baixo desempenho, problemas de escalabilidade e interrupções do sistema.

Existem diversas ferramentas de monitoramento de recursos disponíveis, tanto para ambientes de servidor quanto de nuvem. Algumas das principais funcionalidades incluem o monitoramento em tempo real dos recursos, alertas configuráveis, relatórios históricos e análise de dados. Além disso, algumas ferramentas de monitoramento de recursos também fornecem recursos de gerenciamento de capacidade, como a previsão do uso futuro de recursos e a otimização de recursos. O monitoramento de recursos é essencial para garantir que os sistemas e aplicativos estejam sempre disponíveis e funcionando de maneira eficiente. Ele permite aos administradores da infraestrutura tomar decisões informadas sobre como alocar recursos, planejar a escalabilidade e tomar medidas para resolver problemas de desempenho antes que eles afetem os usuários finais.

### 3.3 Os Problemas ao Monitorar Contêineres

Apesar dos contêineres serem amplamente utilizados para a implantação de aplicativos modernos, o survey da CNCF de 2018 [2], indicou que 34% dos entrevistados listaram o monitoramento como um dos maiores obstáculos ao uso de contêineres. Em comparação com uma solução de monitoramento convencional para infraestrutura virtualizada, a adoção de um sistema de

monitoramento de contêiner apresenta suas próprias dificuldades. Essas dificuldades incluem:

#### 3.3.1 Contêineres são efêmeros

Eles podem ser provisionados rapidamente e destruídos com a mesma rapidez. Esse comportamento é uma das principais vantagens de usá-los, mas pode ser difícil acompanhar as alterações, especialmente em sistemas complexos com alta rotatividade.

#### 3.3.2 Contêineres compartilham recursos

Recursos como memória e CPU são compartilhados, dificultando o monitoramento do consumo de recursos no host físico, o que dificulta a obtenção de uma boa indicação do desempenho do contêiner ou da integridade do aplicativo.

#### 3.3.3 Ferramental Insuficiente

As plataformas de monitoramento tradicionais, mesmo aquelas adequadas para ambientes virtualizados, podem não fornecer informações suficientes sobre métricas, logs e rastreamentos necessários para monitorar e solucionar problemas de integridade e desempenho do contêiner.

## 4. METODOLOGIA

A seguir são definidas as ferramentas avaliadas por esse trabalho e especificada a metodologia utilizada, bem como o detalhamento e execução de cada parte individual.

### 4.1 Ferramentas

Nesse trabalho foram consideradas duas ferramentas utilizadas no monitoramento de contêineres Docker: Prometheus e NetData. A escolha dessas ferramentas se deu levando em consideração o relatório anual da CNCF do ano de 2022 [4]. A seguir, é dada uma definição geral sobre cada uma delas:

- Prometheus: O kit de ferramentas Prometheus foi desenvolvido inicialmente no SoundCloud e consiste em ferramentas para monitorar e alertar sistemas. Desde a sua criação em 2012, várias empresas e organizações adotaram o Prometheus, e o projeto possui uma comunidade ativa de desenvolvedores e usuários. Atualmente, é um projeto autônomo com código aberto que é mantido independentemente de qualquer empresa.
- NetData: Netdata é uma ferramenta de monitoramento em tempo real para sistemas, aplicativos e infraestrutura, incluindo contêineres Docker. Ele oferece uma ampla gama de métricas de performance, incluindo CPU, memória, disco, rede e I/O, bem como métricas de aplicativo e serviço, como número de requisições HTTP e taxa de erro.

### 4.2 Especificação

Neste trabalho optou-se por avaliar qualitativamente um conjunto de cinco pontos específicos nas duas ferramentas avaliadas. São eles:

- Monitoramento em tempo real: A ferramenta deve ser capaz de processar dados em tempo real afim de sinalizar eventuais anomalias e/ou erros no sistema.
- Indicador base de performance: A ferramenta deve ou identificar ou permitir que o usuário defina indicadores bases de performance para fins comparativos com o estado do sistema no momento da execução.

- Monitoramento de performance de rede: A ferramenta deve ser capaz de auxiliar na resolução de problemas referentes a rede monitorando métricas diretamente ligadas à qualidade do serviço pelo usuário final.
- Visualização de dados: A ferramenta deve apresentar um panorama do sistema e das métricas sendo monitoradas através gráficos e dashboards.
- Alerta: A ferramenta deve prover pelo menos uma forma de notificação, ao time ou responsável pela infraestrutura, sobre eventuais problemas/falhas em algum aspecto do sistema.

Para avaliar as ferramentas, será utilizada uma máquina virtual na AWS EC2 instances i3en.large com 2 vCPU, 16 GB RAM e 1 x 1250 GB NVMe SSD para a realização de experimentos. Tais experimentos foram realizados da seguinte forma: Foi utilizada uma simples API escrita em Golang que recebe requisições e realiza computações complexas. Essa API foi containerizada via docker-compose e foi submetida a testes de carga via script, também escrito em Golang, que realizou requisições para a API, afim de simular uma interação real. O experimento foi executado individualmente para cada uma das ferramentas sendo avaliadas nesse trabalho, onde cada execução foi realizada em uma máquina virtual completamente nova.

## 5. RESULTADOS

Nessa seção serão apresentadas as avaliações das ferramentas sendo consideradas nesse trabalho, levando em consideração os critérios do experimento bem como os pontos de avaliação descritos na seção anterior.

### 5.1 Prometheus

- Monitoramento em tempo real: O Prometheus é capaz de coletar e processar dados em tempo real, tornando-o uma ferramenta eficaz para monitoramento de sistemas em produção. Ele oferece suporte a vários tipos de fontes de dados, incluindo APIs REST ou via scripts, além de diversos outros exportadores de métricas, permitindo uma grande variedade de dados sobre todo o sistema.
- Indicador base de performance: O Prometheus permite que o usuário defina indicadores baseados em performance, como CPU, memória e uso de disco. Além disso, ele fornece uma ampla gama de métricas pré-definidas, incluindo métricas de nó, contêiner e aplicativo, que podem ser utilizadas para avaliar o desempenho do sistema ao longo da execução do serviço.
- Monitoramento de performance de rede: O Prometheus oferece suporte ao monitoramento de métricas relacionadas a performance de rede, como taxa de transferência de dados, latência e uso de largura de banda, ajudando assim a identificar problemas de rede e a resolvê-los rapidamente, garantindo a qualidade do serviço prestado ao usuário final.
- Visualização de dados: O Prometheus permite a visualização dos dados coletados através de gráficos. Ele fornece uma interface de usuário pouco intuitiva e que pode ser consideravelmente complexa de navegar, a depender do nível de experiência de quem usa. A interface visual do Prometheus permite navegar pelos

diferentes gráficos e visualizar as métricas coletadas nos contêineres. Além disso, ele oferece integração com ferramentas de visualização de dados externas, como Grafana, que podem ser usadas para criar painéis customizados. Devido a sua modesta e não tão intuitiva interface visual, o Prometheus acaba sendo bem mais utilizado em conjunto com o grafana, onde esses aspectos são significativamente mais robustos e intuitivos.

- Alerta: O Prometheus fornece uma ampla gama de opções de alerta, que permitem que o time ou o responsável pela infraestrutura sejam notificados sobre problemas ou falhas no sistema. Ele suporta a integração com serviços externos, como o PagerDuty, que permitem a criação de alertas personalizados e a configuração de notificações por email, SMS e outras ferramentas para notificação. Além disso, ele oferece uma API de alerta, que permite a integração com outras ferramentas de monitoramento e gerenciamento de incidentes.

### 5.2 NetData

- Monitoramento em tempo real: O NetData é capaz de coletar e processar dados em intervalos de tempo curtos, possibilitando assim, a sinalização de eventuais anomalias e erros no sistema.
- Indicador base de performance: O NetData fornece uma ampla gama de indicadores base de performance para métricas como CPU, memória, disco, rede e outros, permitindo que o usuário defina seus próprios indicadores base.
- Monitoramento de performance de rede: O NetData oferece monitoramento detalhado de performance de rede. Além da vasta quantidade de métricas de performance de rede, o NetData possui também indicadores de qualidade de serviço, o que pode ser muito útil e intuitivo a depender dos casos de uso.
- Visualização de dados: O NetData apresenta uma visualização completa do sistema e das métricas monitoradas através de gráficos interativos e dashboards altamente personalizáveis, de forma própria e automática. Com painéis altamente intuitivos e uma gama de gráficos e dashboards previamente configurados, o monitoramento de contêineres utilizando o NetData acaba se tornando algo bem mais fácil e agradável.
- Alerta: O NetData fornece vários mecanismos de alerta, incluindo notificações por e-mail e mensagens via Slack. Isso garante que o time responsável pela infraestrutura seja notificado rapidamente sobre quaisquer problemas ou falhas no sistema.

## 6. DISCUSSÃO

Após a avaliação das duas ferramentas, o que se verificou foi que o NetData, apesar de exigir a criação de uma conta para poder utilizar, é extremamente fácil de instalar e configurar (caso seja necessário, o que provavelmente ocorrerá somente em casos muito específicos), possui uma UI nativa mais robusta do que a do Prometheus. Além disso, o Netdata coleta uma ampla variedade de métricas, incluindo dados de uso de recursos, desempenho de



processos e tráfego de rede. No entanto, o Netdata não oferece a mesma flexibilidade de personalização de alertas que o Prometheus oferece, e pode ser menos escalável para lidar com grandes quantidades de dados e, por vezes, consumir considerável quantidade de recursos em comparação ao Prometheus.

Por outro lado, o sistema de alerta do Prometheus e sua integração com outras ferramentas de monitoramento e gerenciamento de eventos são os pontos fortes comparados ao concorrente, tornando-o uma solução ideal para grandes organizações. Ele é altamente escalável e permite ao usuário definir regras de alerta personalizadas para eventos críticos. Todavia, o Prometheus pode ser difícil de instalar e configurar, especialmente para usuários sem conhecimento técnico. Sua interface de visualização nativa também é pouco intuitiva o que acaba dificultando ainda mais a experiência do usuário. Isso já não é tão verdade quando o Prometheus é integrado com outras ferramentas, como o Grafana, o que não foi o caso nesse trabalho.

O quadro a seguir avalia o NetData e o Prometheus considerando os pontos usados para avaliação:

Critério	Prometheus	NetData
Monitoramento em tempo real	Ótimo	Ótimo
Indicador base de performance	Bom	Ótimo
Monitoramento de performance de rede	Possui limitações em relação ao NetData. Bom	Ótimo
Visualização de dados	UI nativa mais simples e menos intuitiva do que a do NetData. Regular	Ótimo
Alerta	Excelente	Ótimo

## 7. EXPERIÊNCIAS E LIÇÕES APRENDIDAS

Inicialmente optou-se por criar uma máquina virtual local utilizando Vagrant e Virtual Box para o provisionamento, porém a limitação de hardware fez com que fosse necessário recorrer a um cloud provider, no caso, AWS. Houve uma demanda de esforço na pesquisa por preços e na busca por adaptação para trabalhar com um ambiente de cloud.

Houve também uma demanda de esforço considerável para filtrar informações de tantos relatórios técnicos.

A falta de prática na utilização do Docker Compose também foi um fator exaustivo, sobretudo por causa de erros não tão intuitivos durante a criação dos arquivos compose.

## 8. LIMITAÇÕES E TRABALHOS FUTUROS

Esse trabalho considerou somente duas tecnologias de monitoramento de contêineres citadas no relatório anual da CNCF de 2022 [4], além disso, nos experimentos realizados, somente um contêiner dedicado a API utilizada foi gerado. Assim sugere-se para realização em trabalhos futuros:

- Um comparativo entre as ferramentas avaliadas nesse trabalho e outras ferramentas.
- Um comparativo contemplando ferramentas que não foram contempladas nesse trabalho.
- Um comparativo entre ferramentas unicamente de arquitetura distribuída.
- A repetição desse comparativo porém considerando outros pontos de comparação entre as ferramentas.

Outro comparativo possível seria um entre ferramentas de monitoramento de contêineres Docker que se utilizam de machine learning. Muitas ferramentas de monitoramento incorporam a capacidade de aprender qual é o indicador base de performance e reconhecer comportamentos anômalos nos dados quando eles surgem. Essas ferramentas ainda podem se adaptar às tendências e sazonalidades das métricas para monitorar continuamente a natureza dinâmica do comportamento do sistema e da aplicação e se ajustar automaticamente a situações como picos de utilização.

## 9. AGRADECIMENTOS

Gostaria de agradecer a Deus. A minha família que sempre me motivou e me ajudou desde o primeiro dia de graduação. Aos amigos que fiz durante a graduação e que sempre estiveram ao meu lado. Ao professor Reinaldo pela solicitude e pela compreensão durante os últimos dois semestres, muito obrigado professor. E por fim, a mim mesmo.

## 10. REFERÊNCIAS

- [1] Cloud Native Computing Foundation (CNCF) 2021 Survey. Link: [https://www.cncf.io/wp-content/uploads/2022/02/CNCF-AR\\_FINAL-edits-15.2.21.pdf](https://www.cncf.io/wp-content/uploads/2022/02/CNCF-AR_FINAL-edits-15.2.21.pdf), página 5, acesso: 15 de dezembro de 2022.
- [2] CNCF Survey: Use of cloud native technologies in production has grown over 200%. Link: <https://www.cncf.io/blog/2018/08/29/cncf-survey-use-of-cloud-native-technologies-in-production-has-grown-over-200-percent/>, acesso: 03 de janeiro de 2023.
- [3] Use container to build, share and run your applications. Link: <https://www.docker.com/resources/what-container/>, acesso: 07 de janeiro de 2023.
- [4] CNCF Annual Report 2022: Building for the road ahead. Link: <https://www.cncf.io/reports/cncf-annual-report-2022/>, acesso: 05 de janeiro de 2023.
- [5] Homepage oficial do NETDATA. <https://www.netdata.cloud/>.
- [6] Homepage oficial do Prometheus. <https://prometheus.io/>.

- [7] Repositório oficial do cAdvisor no Github. <https://github.com/google/cadvisor>.
- [8] SILVA, Tainah Emmanuele. Um processo para administração de serviços em contêineres Docker. 2019. 12f. (Trabalho de Conclusão de Curso - Artigo) – Curso de Bacharelado em Ciência da Computação, Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Paraíba, Brasil, 2019. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/20778>
- [9] V. Pfeifer, W. F. Passini, W. F. Dorante, I. R. Guilherme and F. J. Affonso, A multi-agent approach to monitor and manage container-based distributed systems, in IEEE Latin America Transactions, vol. 20, no. 1, pp. 82-91, Jan. 2022, doi: 10.1109/TLA.2022.9662176

---

## Sobre os autores:

**Lucas Chaves Salustiano**, graduando em Ciência da Computação pela Universidade Federal de Campina Grande (UFCG). Tem interesse em teoria da computação, engenharia de software, arquitetura de software, administração de sistemas e de infraestrutura. Atualmente atua como desenvolvedor de software.

**Prof. Dr. Reinaldo César de Moraes Gomes**, graduado em Tecnologia em Telemática pelo Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), mestrado e doutorado em Ciências da Computação pela Universidade Federal de Pernambuco (UFPE). Sua experiência tem ênfase em Redes de Computadores e Sistemas Distribuídos. Suas principais áreas de atuação são: Negotiation, Self-configuration, Dynamic Networks, Policies, Inter-Domain Communication. Já teve experiência como professor do Instituto Federal de Educação, Ciência e Tecnologia de Alagoas (IFAL).