

**CENTRO UNIVERSITÁRIO FEI**

**MURILO DARCE BORGES SILVA**

**RODRIGO SIMÕES RUY**

**IDENTIFICAÇÃO DE DIFERENÇAS DE DESEMPENHO ENTRE SISTEMAS  
ROBÓTICOS SIMULADOS COM E SEM SOFTWARE CONTEINERIZADO,  
UTILIZANDO SIMULADOR GAZEBO, ROS 2 E DOCKER.**

São Bernardo do Campo

2025

MURILO DARCE BORGES SILVA  
RODRIGO SIMÕES RUY

**IDENTIFICAÇÃO DE DIFERENÇAS DE DESEMPENHO ENTRE SISTEMAS  
ROBÓTICOS SIMULADOS COM E SEM SOFTWARE CONTEINERIZADO,  
UTILIZANDO SIMULADOR GAZEBO, ROS 2 E DOCKER.**

Trabalho de Conclusão de Curso apresentado ao  
Centro Universitário FEI, como parte dos requisitos  
necessários para obtenção do título de Bacharel em  
Ciência da Computação. Orientado pelo Prof. Dr.  
Leonardo Anjoletto Ferreira.

São Bernardo do Campo

2025

MURILO DARCE BORGES SILVA  
RODRIGO SIMÕES RUY

**IDENTIFICAÇÃO DE DIFERENÇAS DE DESEMPENHO ENTRE SISTEMAS  
ROBÓTICOS SIMULADOS COM E SEM SOFTWARE CONTEINERIZADO,  
UTILIZANDO SIMULADOR GAZEBO, ROS 2 E DOCKER.**

Trabalho de Conclusão de Curso apresentado ao  
Centro Universitário FEI, como parte dos requisitos  
necessários para obtenção do título de Bacharel em  
Ciência da Computação.

Comissão julgadora

Prof. Dr. Leonardo Anjoletto Ferreira

Prof. Dr. Plinio Thomaz Aquino Junior

Prof. Dr. Fagner de Assis Moura Pimentel

São Bernardo do Campo

01/12/2025

## **AGRADECIMENTOS**

Agradecemos ao professor Fagner Pimentel pelo modelo para simulação do labirinto que existe na K4-04, que foi de suma importância para o desenvolvimento do projeto.

## SUMÁRIO

<b>1</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>4</b>
1.1	Bare-Metal vs. Hypervisors and Containers: Performance Evaluation of Virtualization Technologies for Software-Defined Vehicles . . . . .	4
1.2	Docker Performance Evaluation across Operating Systems . . . . .	4
1.3	DA Containerized Microservice Architecture for a ROS 2 Autonomous Driving Software: An End-to-End Latency Evaluation . . . . .	5
	REFERÊNCIAS . . . . .	6

## 1 TRABALHOS RELACIONADOS

### DESENVOLVER, COLOCAR METODOLOGIA DOS PROJETOS

#### 1.1 BARE-METAL VS. HYPERVISORS AND CONTAINERS: PERFORMANCE EVALUATION OF VIRTUALIZATION TECHNOLOGIES FOR SOFTWARE-DEFINED VEHICLES

O projeto de Wen et. al. (2023), auxilia com relação ao entendimento da containerização em sistemas embarcados e também com relação ao seu desempenho. Este projeto No artigo, é detalhada a utilização de diferentes formas de containerização e seu efeito no desempenho em diferentes tipos de hardware. Foram realizados testes gerais que envolviam CPU, memória, rede e disco, em três ambientes diferentes: máquina virtual, contêiner e um contêiner dentro de uma máquina virtual. Com isso, foi concluído que as máquinas virtuais e os contêineres possuem um desempenho semelhante ao bare-metal (servidor físico que é de uso exclusivo para apenas um cliente, sem a camada de virtualização que fica entre o hardware e o sistema operacional), onde entre a CPU, rede e memória, sofria uma perda de no máximo 5% enquanto no disco a diferença era de até 35%. Foi observado que o Docker e a KVM (máquina virtual baseada no Kernel) foram 5 a 10% mais lentos que o bare-metal, com o Docker sendo mais lento ainda na primeira inicialização, mas levando a concluir que containerização e virtualização podem ser utilizados em aplicações para automóveis. (WEN et al., 2023).

#### 1.2 DOCKER PERFORMANCE EVALUATION ACROSS OPERATING SYSTEMS

O trabalho de Sobieraj e Kotyński (2024), seu projeto auxilia no entendimento dos conceitos de avaliação do Docker com relação a outros sistemas operacionais. Este estudo realiza testes em que o Docker é executado em diversos sistemas operacionais, para isso foi utilizado um MacBook Pro 13, com uma CPU da Intel (Intel i5-8257u @ 1.40 GHz) como plataforma do hardware para a comparação de desempenho, 16 GB LPDDR3 2133 Mhz de memória RAM e um armazenamento de 256 GB NVME SSD Para verificar a diferença entre os sistemas operacionais, foram utilizados sistemas operacionais recém-instalados que eram MacOS Ventura, Ubuntu 22.04 e Windows 10 rodando em um MacBook Pro 13 MacOS Ventura 13.5.1, Ubuntu 22.04-6.4.8-t2-jammy e Windows 10 22H2. Os testes consistiam em estressar o Docker com

relação à CPU, rede e na resiliência do mesmo. Após realizar os testes, foi observado que o sistema operacional possui uma importante influência sobre o desempenho presente no container Docker. Alguns possuíam benefícios em relação a outros em uma determinada categoria. O macOS se destacou com relação aos dados obtidos nas configurações utilizadas nos sistemas docker, não sofrendo grandes perdas de desempenho, se mostrando extremamente versátil, o Linux se mostrou mais eficiente quanto às aplicações que raramente utilizam escrita no disco, se mostrando uma escolha melhor que o MacOS com relação a bancos de dados em memória, cache e entre outros, pois por não necessitarem de tanta escrita, essas aplicações se beneficiam mais quando executadas em um container com Linux, o Windows acabou não se beneficiando tanto quanto os outros, a não ser pela taxa de transferência de rede entre os contêineres. Assim como o Linux, o Windows possui problemas com a velocidade em que a escrita é feita e com isso. Este artigo auxilia no entendimento com relação aos tipos de testes que podem ser realizados para analisar o desempenho do Docker, auxiliando de uma maneira que possa ser um pontapé inicial para o desenvolvimento dos testes com o Docker, e como medir o desempenho de um contêiner. (SOBIERAJ; KOTYŃSKI, 2024)

### 1.3 DA CONTAINERIZED MICROSERVICE ARCHITECTURE FOR A ROS 2 AUTONOMOUS DRIVING SOFTWARE: AN END-TO-END LATENCY EVALUATION

Conforme Wen et. al. (2024), o artigo aborda a arquitetura baseada em microserviços para sistemas automotivos. Cada serviço foi realizado em contêineres separados, pois os testes realizados identificaram que este método é viável e acaba por melhorar a latência existente em sistemas operacionais Linux sem contêineres, que obteve uma latência de 5 a 8% end-to-end, além de reduzir a latência máxima, o que mostra a vantagem no uso de containerização para os sistemas automotivos em tempo real. Foi concluído que o ROS 2, utilizado para avaliar a arquitetura de microserviços para uma aplicação real de direção autônoma, foi de extrema importância por conta de sua arquitetura distribuída que é baseada em nós e possui comunicação DDS, o que levou ao isolamento das funções do veículo e facilitou a migração para contêineres. Na containerização, o ROS 2 perde um pouco de seu desempenho, mas em aplicações complexas como o Autoware, a mesma acaba por melhorar, reduzindo o uso de CPU e memória. Este artigo auxilia no entendimento do uso da containerização com relação ao ROS 2, seus problemas e seus acertos. (WEN et al., 2024).

## REFERÊNCIAS

SOBIERAJ, M.; KOTYŃSKI, D. Docker performance evaluation across operating systems. **Applied Sciences**, v. 14, n. 15, 2024. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/14/15/6672>>.

WEN, L. et al. **A Containerized Microservice Architecture for a ROS 2 Autonomous Driving Software: An End-to-End Latency Evaluation**. 2024. Disponível em: <<https://arxiv.org/abs/2404.12683>>.

\_\_\_\_\_. Bare-metal vs. hypervisors and containers: Performance evaluation of virtualization technologies for software-defined vehicles. In: . [s.n.], 2023. Disponível em: <[https://www.researchgate.net/publication/372496789\\_Bare-Metal\\_vs\\_Hypervisors\\_and\\_Containers\\_Performance\\_Evaluation\\_of\\_Virtualization\\_Technologies\\_for\\_Software-Defined\\_Vehicles](https://www.researchgate.net/publication/372496789_Bare-Metal_vs_Hypervisors_and_Containers_Performance_Evaluation_of_Virtualization_Technologies_for_Software-Defined_Vehicles)>.