

## Containerização: Panorama tecnológico de sua utilização por alunos egressos do Curso de Tecnologia em Sistemas para Internet da UTFPR - Campus Guarapuava

### Containerization: Technological overview of its use by students graduating from the Internet Systems Technology Course at UTFPR - Guarapuava Campus

Gustavo Alexandre Dias<sup>1</sup>, Sediane Carmem Lunardi Hernandez<sup>2</sup>

#### RESUMO

Na informática, devido às mudanças do mercado, novas metodologias e tecnologias surgem rapidamente. Isso requer que os profissionais da área de tecnologia mantenham-se atualizados. Sendo assim a universidade possui um papel fundamental neste processo. A Universidade Tecnológica Federal do Paraná (UTFPR) campus Guarapuava oferece dois cursos de tecnologia, sendo um deles o curso Tecnologia em Sistemas para Internet (TSI). O curso busca formar profissionais para atuar nas áreas de desenvolvimento web, administração de serviços, projeto de redes de computadores e sistemas para internet. Diante disso, elaborou-se um projeto que busca realizar uma pesquisa com os alunos egressos do curso de TSI visando obter um panorama geral das tecnologias ligadas à área de infraestrutura que são utilizadas nas empresas em que os alunos trabalham. A pesquisa demonstrou que os tópicos de estudo atuais que já vinham sendo trabalhados no curso e incorporados na nova matriz, como por exemplo, Docker, Kubernetes, Sistemas de Controle de Versão são amplamente utilizados nas empresas pelos alunos, mostrando assim, que a nova matriz do curso está alinhada ao mercado de trabalho.

**PALAVRAS-CHAVE:** Docker; Sistema de Controle de Versão; Curso de Tecnologia em Sistemas para Internet.

#### ABSTRACT

In the field of computer science, due to market changes, new methodologies and technologies emerge rapidly. This requires technology professionals to stay updated. Therefore, the university plays a crucial role in this process. The Federal University of Technology of Paraná (UTFPR), Guarapuava campus, offers two technology programs, one of which is the Technology in Internet Systems (TSI) program. The program aims to educate professionals in the areas of web development, service administration, computer network design, and internet systems. In light of this, a project was developed to conduct research with TSI program graduates, seeking to obtain a comprehensive overview of the technologies related to infrastructure used by the companies where the students work. The research demonstrated that current study topics that were already integrated into the curriculum, such as Docker, Kubernetes, and Version Control Systems, are widely employed by the students in the workplace. This shows that the new curriculum aligns with the demands of the job market.

**KEYWORDS:** Docker; Version Control System; Internet Systems Technology Course.

#### INTRODUÇÃO

Na área de informática, novas metodologias de desenvolvimento, linguagens de programação, tecnologias de suporte surgem rapidamente devido às exigências do mercado de trabalho. Assim, o profissional de Tecnologia da Informação (TI) precisa estar sempre disposto a estar atualizando seu conhecimento. Neste sentido, a Universidade possui um papel fundamental nesse processo, produzindo conhecimentos,

<sup>1</sup> Bolsista do Programa de Iniciação Científica e Tecnológica. Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil. E-mail: [gustavodias.2000@alunos.utfpr.edu.br](mailto:gustavodias.2000@alunos.utfpr.edu.br), ID Lattes: 7030714586062825.

<sup>2</sup> Docente do Curso de Tecnologia em Sistemas para Internet. Universidade Tecnológica Federal do Paraná, Guarapuava, Paraná, Brasil. E-mail: [sedianec@utfpr.edu.br](mailto:sedianec@utfpr.edu.br), ID Lattes: 2421420688966710.

técnicas e tecnologias por meio da pesquisa; promovendo o ensino de qualidade e realizando a extensão universitária. Na Universidade Tecnológica Federal do Paraná (UTFPR), isso não é diferente. A UTFPR tem como missão “Desenvolver a educação tecnológica de excelência por meio do ensino, da pesquisa e da extensão, interagindo de forma ética, sustentável, produtiva e inovadora com a comunidade para o avanço do conhecimento e da sociedade.” (PDI 2018-2022). A educação tecnológica de excelência é realizada em seus diversos cursos de pós-graduação e graduação como os cursos de bacharelado e tecnologia.

O Campus Guarapuava da UTFPR possui dois cursos de tecnologia, sendo um deles o curso de Tecnologia em Manutenção Industrial (TMI) e Tecnologia em Sistemas para Internet (TSI). O curso de TSI iniciou suas atividades no Campus Guarapuava em março de 2011. O curso já passou por uma mudança de matriz e iniciou a implantação de uma matriz por competências no primeiro semestre de 2023.

Para desenvolver competências, é fundamental abordar situações e problemas reais (WITTER; ZABALA; ARNAU, 2010), que ajudam a identificar os conhecimentos e habilidades necessários para uma profissão. Nesse contexto, analisar as tecnologias usadas por empresas com egressos do curso TSI é essencial. Essa pesquisa visa fornecer uma visão abrangente das tecnologias, como Docker, máquinas virtuais, computação em nuvem e controle de versão de projetos de software, para alinhar os tópicos de estudo da linha de infraestrutura na nova matriz de competências do curso de TSI. O artigo é dividido como segue. Na seção 1 apresenta-se o referencial teórico do trabalho. A seção 2 trata dos materiais e métodos e na seção 3 são descritos os resultados do trabalho.

## DESENVOLVIMENTO

Essa seção apresenta o referencial teórico que embasou a pesquisa realizada.

### Máquinas virtuais X Containers

A virtualização de computadores ou servidores trata do “processo de, em uma mesma máquina física, simular dois ou mais computadores, possibilitando que cada um contenha uma parte do software e funcione de forma totalmente independente um do outro” (MONTEIRO et al., 2021, p. 102). Com isso, o sistema operacional se responsabiliza pela comunicação entre as máquinas virtuais. Entretanto, o gerenciamento de muitas máquinas virtuais pode ser complexo. Assim, o conceito de container surge como uma maneira mais simples e leve de trabalhar do que com máquinas virtuais.

Uma distinção relevante é que os containers, em comparação com as máquinas virtuais, não precisam carregar um sistema operacional completo e compartilham do mesmo kernel do *host*, tornando-os mais escaláveis (NICKOLOFF, 2016). Containers podem ser vistos como máquinas virtuais leves (ANDERSON, 2015), isolando o software de seu ambiente independente das variações de ambiente de desenvolvimento ou teste.

Docker é um container que possibilita aos desenvolvedores empacotar projetos e suas dependências em um único pacote padronizado, resolvendo problemas de implantação e permitindo a execução em qualquer máquina (SILVA, 2016). O Docker

utiliza imagens como base para criar containers, que podem ser iniciados a partir de imagens existentes ou criados a partir de Dockerfiles personalizados (MATTHIAS; KANE, 2016). As imagens do Docker são pacotes autônomos que contém tudo o necessário para executar uma aplicação, organizadas em camadas definidas por instruções no Dockerfile (DOCKER, 2022). Além disso, o Docker utiliza técnicas de armazenamento e drivers para gerenciar dados e arquivos usados pelos containers (KOHLI et al., 2017).

### Sistema de Controle de Versão

Um sistema de controle de versão, também conhecido como VCS (*Version Control System*) possui “como finalidade gerenciar várias versões no desenvolvimento de um documento qualquer.” (MONTEIRO et. al., 2021, p. 23).

Os sistemas de controle de versão são compostos por duas partes principais, (CHACON; STRAUB, 2014): o **repositório**, que atua como um servidor centralizado, mantendo o registro completo das alterações no projeto, e a **estação de trabalho** individual de cada desenvolvedor, que possui uma cópia dos arquivos do projeto conectada ao servidor. Essa estrutura facilita a detecção de modificações e sincronismos com os arquivos de projeto com o *host*, além de manter cada estação de trabalho independente e isolada.

O Git é um sistema de controle de versão de código-fonte distribuído e de código aberto, criado por Linus Torvalds para o desenvolvimento do Kernel do Linux (CHACON; STRAUB, 2014). Suas características incluem organização do código pela equipe de desenvolvimento, criação de históricos de funcionalidades, backup do projeto, restauração de código e criação de novas versões integradas de um mesmo projeto. Para gerenciar repositórios na web, muitos desenvolvedores optam por utilizar o GitHub ou o Bitbucket em suas próprias máquinas (CHACON; STRAUB, 2014). O GitHub é uma plataforma central de colaboração para desenvolvedores e projetos, utilizada por projetos de código aberto para hospedagem de repositórios Git, rastreamento de problemas e revisão de código.

### METODOLOGIA

Para o desenvolvimento deste projeto utilizou-se o método de pesquisa survey. O método de pesquisa survey pode ser definido como “a obtenção de dados, ações ou opiniões de determinado grupo de pessoas, indicado como representante de uma população-alvo, por meio de um instrumento de pesquisa, normalmente um questionário” (TANUR apud PINSONNEAULT; KRAEMER, 1993). A pesquisa descritiva descreveu o uso de tecnologias, metodologias e ferramentas pelos alunos egressos do curso de TSI da UTFPR - Campus Guarapuava, examinando suas inter-relações. A coleta de dados ocorreu em um único momento com uma amostra probabilística de 23 alunos, de um total de 50 egressos. Foi usado um questionário no Google Forms como instrumento de pesquisa, em que as perguntas contidas nesse questionário abordaram questões voltadas ao tema de estudo, ou seja, container, sistemas de controle de versões Docker, máquinas virtuais e conceitos de infraestrutura de computadores. As perguntas aplicadas podem ser encontradas no link <https://forms.gle/pNLUmNjFwqM14GEw5>. O

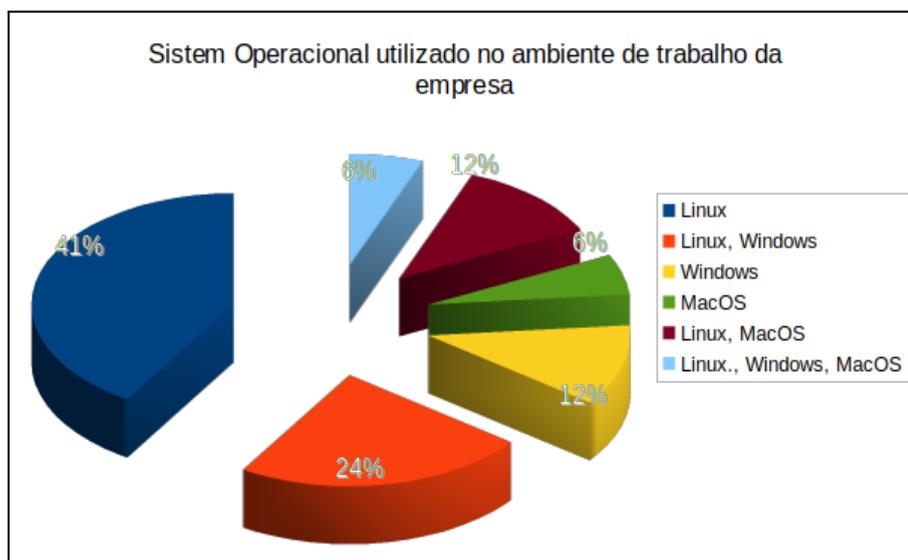
link enviado aos participantes inclui um termo de confidencialidade assinado pelo pesquisador, garantindo o anonimato dos dados dos participantes. Além da aplicação do questionário, desenvolveu uma revisão bibliográfica sobre containerização, virtualização e controle de versões.

## RESULTADOS

A pesquisa mostrou que 74% dos entrevistados estão trabalhando em empresas envolvidas com o desenvolvimento de software e 26% não. Dos que trabalham em empresas de desenvolvimento de software, que será o enfoque dos dados apresentados, 65% trabalham em empresas multinacionais e 35% em empresas nacionais, sendo que 76% trabalham de forma remota e 24% de forma híbrida (remota e presencial).

Em relação ao uso do sistema operacional utilizado no ambiente de trabalho da empresa, no gráfico apresentado pela Figura 1, o Linux é utilizado por 82% dos respondentes. De todos os entrevistados, embora utilizem outro SO, 92% responderam que possuem conhecimento intermediário em Linux e 7% conhecimento avançado. Em relação a problemas de compatibilidade entre sistemas operacionais ao desenvolver ou executar projetos, 71% responderam que já tiveram problemas e 29% nunca tiveram.

Figura 1 – Sistema Operacional mais utilizado no ambiente de trabalho

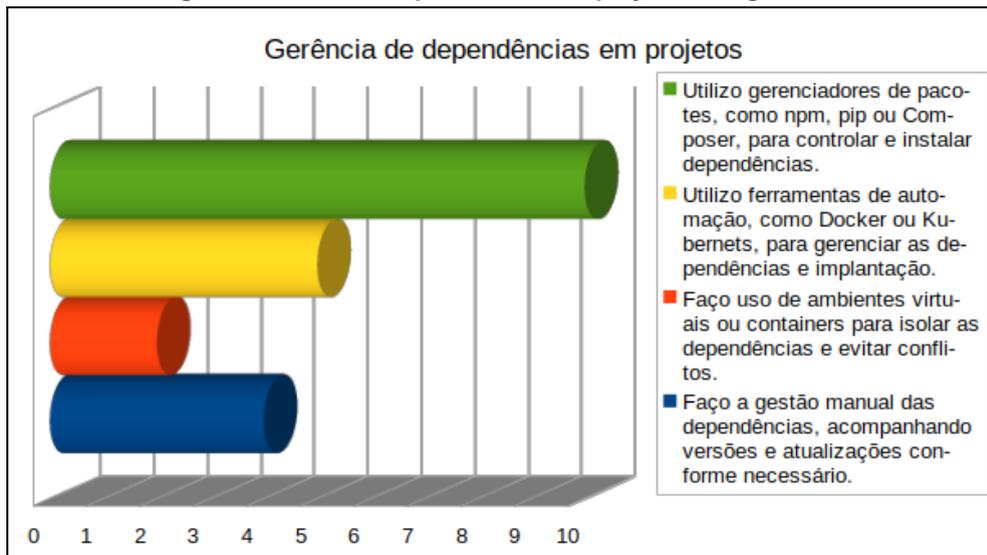


Fonte: Autoria própria.

Quanto às dependências em projetos (Figura 2), a maioria dos entrevistados usa gerenciadores de pacotes para controlar e instalar dependências. Alguns optam por ferramentas de automação, como Docker ou Kubernetes, enquanto outros preferem ambientes virtuais ou containers para isolar dependências e evitar conflitos. Além disso, há quem realize o controle manual. Muitos entrevistados relatam realizar mais de uma atividade simultaneamente. Ainda, sobre o Docker, 88% dos alunos egressos trabalham com o Docker e 94% deles acredita que empacotando as aplicações com o Docker tornando-as portáteis, melhora o processo de *deploy* (implantação) das aplicações.

Destes, todos já utilizaram o Docker Compose, 50% já utilizou Kubernetes, 17% o Docker Swarm e 24% já utilizou o Docker Secrets. Na média realizada, todos consideraram o Docker fácil de ser utilizado. O Docker é executado em ambiente Linux por 76% dos entrevistados e os outros 24% utilizam o Docker em ambiente Windows ou MacOS, divididos igualmente.

Figura 2 – Como as dependências em projetos são gerenciadas



Fonte: Autoria própria.

Cerca de 94% dos egressos utilizam sistemas de controle de versão, destacando benefícios como organização em equipe, aprimoramento de testes automatizados e deploys, facilidade de colaboração em equipe, manutenção de código e integração de ferramentas de entrega contínua. Todos concordam que o GitHub facilita a colaboração em projetos de software. Além disso, 100% das empresas em que os egressos trabalham utilizam serviços em nuvem para hospedar aplicativos e recursos. Dessas empresas, 76% ainda alocam máquinas virtuais em ambiente de nuvem para suportar cargas de trabalho, 29% utilizam uma combinação de serviços em nuvem e recursos locais. Dentre os tipos de nuvem que as aplicações são hospedadas foram elencados Amazon Web Services (AWS), Digital Ocean, Microsoft Azure, Google Cloud Computing (GCP) e Oracle Cloud.

### CONSIDERAÇÕES FINAIS

A análise do exercício profissional dos alunos egressos do Curso de Tecnologia em Sistemas para Internet permitiu realizar uma análise geral sobre as tecnologias e ferramentas utilizadas pelas empresas em que os alunos egressos do curso de TSI estão inseridos. O uso das tecnologias pelos alunos egressos mostra a importância de que essas tecnologias sejam ainda mais incorporadas no Curso de Tecnologia em Sistemas para Internet na linha de infraestrutura da nova matriz.

As empresas adotarem o Linux como sistema operacional confirma a escolha do curso, alinhando-se às práticas do mercado. O uso generalizado de containers Docker

para gerenciar dependências em projetos, tanto no curso quanto no mercado de trabalho, é outro indicativo de alinhamento. Além disso, o uso generalizado da ferramenta Git, presente desde o primeiro período do curso, demonstra que a matriz curricular está em sintonia com as tecnologias utilizadas pelas empresas.

Futuramente, pretende-se estender essa pesquisa para as empresas de software de Guarapuava e região.

### Agradecimentos

A Universidade Tecnológica Federal do Paraná por tornar possível a realização deste trabalho e ao professor Hermano Pereira pela consultoria.

### Conflito de interesse

“Não há conflito de interesse”.

### REFERÊNCIAS

ANDERSON, C. Docker [Software engineering]. **IEEE Software**, vol. 32, no. 3, pp. 102-c3, May-June 2015, doi: 10.1109/MS.2015.62.

CHACON, S.; STRAUB, B. **Pro Git**. Apress, 2a. ed. Disponível por www em: <https://git-scm.com/book/en/v2>. Acesso em 12 de maio de 2023.

DOCKER. **Package Software into Standardized Units for Development, Shipment and Deployment**. Disponível por www em <https://www.docker.com/resources/what-container/>. Acesso em 03 de maio de 2022.

KOHLI, V.; DUA, R.; WOOTEN, J. **Troubleshooting Docker**: Strategically design, troubleshoot, and automate Docker containers from development to deployment. Packt, 2017.

MATTHIAS, K.; KANE, S. P. **Primeiros passos com Docker**: Usando containers em produção. Novatec, 2016.

MONTEIRO, E. R. *et al.* **DevOps**. Porto Alegre: Grupo A, 2021. 9786556901725. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556901725/>. Acesso em: 26 mai. 2022.

NICKOLOFF, J. **Docker in Action**. MANNING, 2016.

PINSONNEAULT, A.; KRAEMER, K. Survey research methodology in management information systems: an assessment. **Journal of management information systems**, v. 10, n. 2, p. 75-105, 1993.

SILVA, W. F. da S. **Aprendendo DOCKER**: Do básico à orquestração de containers. São Paulo: Novatec, 2016.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR. Plano de Desenvolvimento Institucional (**PDI**) 2018-2022. Curitiba, 2017.

WITTER, G. P.; ZABALA, A.; ARNAU, L. **Como aprender e ensinar competências**. Tradução de Carlos Henrique Lucas Lima. Porto Alegre: Artmed, 2010. *Práxis Educativa*, v. 6, n. 1, p. 133-134, 2011.