# VIRTULIZAÇÃO DE SISTEMAS LINUX NO WSL: Um estudo de caso sobre a ferramenta de virtualização desenvolvida pela Microsoft

José Florindo Sgorlon Filho, Vitor Roberto de Souza, Orientador: Me. José Aparecido de Aguiar Viana

e-mail:
jose.sgorlon@fatec.sp.gov.br; vitor.souza48@ fatec.sp.gov.br; jose.viana@fatec.sp.gov.br

***Resumo:*** *Atualmente, muitas soluções em tecnologia da informação utilizam máquinas virtuais para hospedar uma variedade de aplicativos, desde desktops virtuais para empresas até serviços e sites acessados globalmente. O Windows Subsystem for Linux (WSL) é uma tecnologia desenvolvida pela Microsoft que permite que os usuários do Windows executem simultaneamente sistemas operacionais Windows e Linux em seus computadores de forma nativa, sem a necessidade de dual boot. Este trabalho teve como objetivo realizar um estudo de caso sobre o WSL, relacionando suas características com os conceitos teóricos de virtualização de sistemas operacionais e identificando suas diferenças em relação a outras soluções, destacando sua utilidade para desenvolvedores que necessitam acessar ferramentas de ambos os sistemas simultaneamente. Ao final deste estudo, foi possível observar que o WSL, em suas duas versões, é uma solução efetiva e promissora para a virtualização de sistemas operacionais Linux no ambiente Windows de forma rápida e sem a configurações iniciais complexas.*

***Palavras-chave:*** *Subsistema Windows para Linux, Máquinas virtuais, Máquina virtual Linux no Windows, Desenvolvimento de software em máquinas virtuais.*

***Abstract:*** *Currently, many information technology solutions utilize virtual machines to host a variety of applications, ranging from virtual desktops for businesses to globally accessible services and websites. Windows Subsystem for Linux (WSL) is a technology developed by Microsoft that allows Windows users to simultaneously run Windows and Linux operating systems on their computers natively, without the need for dual boot. This work aimed to conduct a case study on WSL, relating its characteristics to theoretical concepts of operating system virtualization and identifying its differences compared to other solutions, highlighting its usefulness for developers who need simultaneous access to tools from both systems. At the end of this study, it was possible to observe that WSL, in its two versions, is an effective and promising solution for the virtualization of Linux operating systems in the Windows environment quickly and without complex initial configurations.*

***Keywords:*** *Windows Subsystem for Linux, Virtual machines, Linux virtual machine on Windows, Software development on virtual machines.*

## 1. Introdução

Atualmente, diversas soluções desenvolvidas na área da tecnologia da informação fazem uso em certa medida das chamadas máquinas virtuais para sua viabilização, estas máquinas podem ser responsáveis por hospedar os mais diversos tipos de soluções que vão desde o acesso a uma área de trabalho virtualizada para cada um dos colaboradores de uma empresa ou a hospedagem de diversos micro serviços e sites acessados por pessoas de todo mundo. Segundo Randal (2020), o desenvolvimento das máquinas virtuais cresceu a partir do conceito de multiprogramação que tinha como foco o compartilhamento seguro de recursos de *hardware* disponíveis em um equipamento e o ampliou para a ideia de que pequenos agrupamentos de recursos compartilhados por uma máquina poderiam ser isolados e passar a ilusão de estar sendo executados em uma outra máquina com um sistema operacional completo.

Outros dois fatores que impulsionaram o desenvolvimento e surgimento das máquinas virtuais segundo Carissimi (2008) foi que primeiro na década de 70 os sistemas computacionais vinham com sistema operacionais próprios e por vezes exclusivos de seus fabricantes. Segundo o autor, o posterior aumento de poder computacional, aumento sistemas distribuídos e onipresença das redes de computadores fez com que a busca por soluções que facilitassem executar aplicações independente o sistema operacional usado aumentasse.

O *Windows Subsystem for Linux* (*WSL*) é uma tecnologia desenvolvida pela *Microsoft* que permite que os usuários executem sistemas operacionais *Linux* em seus computadores com o *Windows* como sistema operacional principal. O *WSL* foi introduzido pela primeira vez em 2017 com o *Windows 10 Fall Creators Update* e tem sido continuamente melhorado desde então. (RAJ, 2017)

Uma das principais vantagens do *WSL* é que ele permite que os usuários acessem e usem ferramentas *Linux*, como o terminal *bash* e o gerenciador de pacotes disponíveis pelo gerenciados *apt*, diretamente no ambiente do *Windows* sem a necessidade de *dual boot*. Essa característica especialmente útil para desenvolvedores que desejam trabalhar com tecnologias baseadas em *Linux*, como aplicativos *web* e sistemas de gerenciamento de banco de dados, sem precisar alternar para um sistema operacional *Linux* separado.

Além disso, o *WSL* permite que os usuários compartilhem arquivos e pastas entre o *Windows* e o *Linux*, o que pode ser útil para tarefas de desenvolvimento e administração de sistemas. O *WSL* também suporta a execução de aplicativos *Linux* em janelas de console, além de aplicativos com interface gráfica, graças à integração do *X11* e *Wayland*.

O *WSL* seria então uma "ponte" para a comunidade de desenvolvedores que permite aos usuários do *Windows* aproveitar o melhor dos dois mundos. Com o *WSL*, a *Microsoft* está oferecendo aos usuários a possibilidade de executar sistemas operacionais *Linux* no *Windows* sem a necessidade de configurações complexas, com bom desempenho e integração entre os sistemas. (MICROSOFT, 2022)

## 2. Justificativa

Uma vez que se torna cada vez mais comum a utilização de máquinas virtuais ou estratégias de conteinerização nos servidores de produção para a hospedagem dos mais diversos serviços e aplicações em TI, é comum que o ambiente no qual os desenvolvedores constroem a aplicação não sejam o mesmo que ela será hospedada. Dentre as possíveis soluções que visam facilitar o processo de desenvolvimento e teste das aplicações está o *WSL* da *Microsoft* que faz uso da virtualização para permitir que o desenvolvedor tenha acesso as ferramentas utilizadas tanto no sistema operacional *Linux* ou *Windows* de forma simultânea.

## 3. Objetivo(s)

Este trabalho teve como objetivo apresentar um estudo de caso sobre o *WSL* desenvolvido pela empresa *Microsoft*, buscando relacionar o seu funcionamento com os conceitos teóricos acerca das técnicas de virtualização de sistemas operacionais, e comparar as versões 1 e 2 do *WSL* como ferramentas para desenvolvedores de *software*.

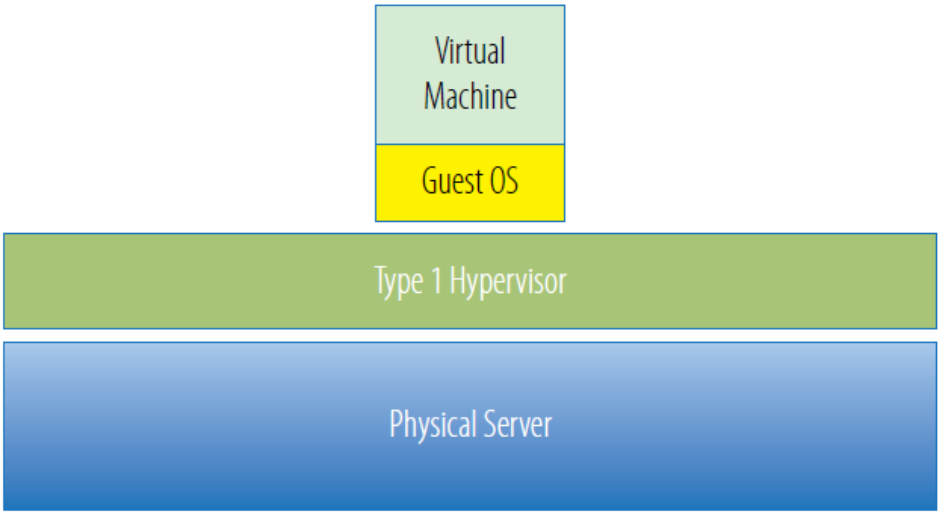
## 4. Fundamentação Teórica

A virtualização é uma técnica que permite a criação de ambientes virtuais independentes do *hardware* físico. Essa técnica possibilita a execução de múltiplos sistemas operacionais em uma única máquina física. A virtualização tem diversas aplicações práticas, como a consolidação de servidores, a criação de ambientes de testes e desenvolvimento, a implantação de infraestrutura de nuvem, entre outras. Existem duas categorias principais de virtualização, cada um com suas particularidades: *hypervisor* tipo 1 ou virtualização de tipo 1 e *hypervisor* tipo 2 ou virtualização de tipo 2.

Portnoy (2016) explica um *hypervisor* como sendo uma camada de software que fica entre o hardware e uma ou mais máquinas virtuais que ele suporta. Suas fusões são bastante simples e se concentram em fornecer um ambiente idêntico ao ambiente físico, fornecer esse ambiente com custo de desempenho mínimo e manter controle completo dos recursos do sistema.

De acordo com Portnoy (2016), os *hypervisors* tipo 1, também conhecidos como virtualização de nível de *hardware*, são executados diretamente no *hardware* do servidor, sem um sistema operacional abaixo dele conforme demonstrado na Figura 1. Por não haver uma camada intermediária entre o *hypervisor* e o *hardware* físico, isso também é conhecido como uma implementação *"bare-metal"* e por não haver intermediários ele pode se comunicar diretamente com os recursos de *hardware* na pilha abaixo dele, tornando-o muito mais eficiente do que o *hypervisor* Tipo 2.

**Figura 1** *hypervisor* tipo 1

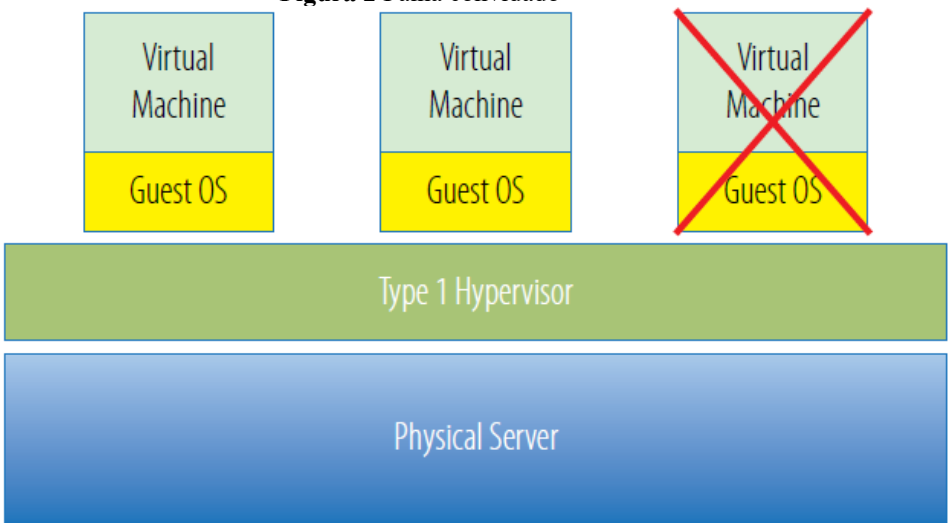



Fonte: PORTNOY, 2016

Além de ter melhores características de desempenho, Portnoy (2016) afirma que os *hypervisors* tipo 1 também são considerados mais seguros do que os *hypervisors* tipo 2. As operações dos convidados são transferidas e, como tal, um convidado não pode afetar o *hypervisor* em que está sendo executado. Uma máquina virtual pode danificar apenas a si mesma, causando uma falha isolada do convidado, mas esse evento não escapa dos limites do

contêiner da máquina virtual conforme demonstrado na Figura 2. Outros convidados continuam processando, e o *hypervisor* também não é afetado. Um convidado malicioso, onde o código está tentando deliberadamente interferir com o *hypervisor* ou com os outros convidados, não conseguiria fazê-lo.

**Figura 2** Falha convidado

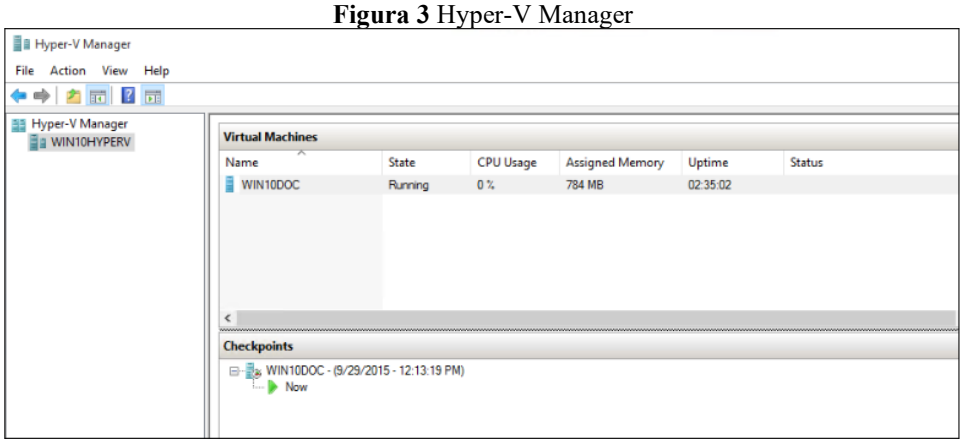



Fonte: PORTNOY, 2016

Portnoy (2016) destaca que menos recursos de processamento são necessários para um *hypervisor* tipo 1, o que significa que mais máquinas virtuais podem ser executadas em cada *host*. Do ponto de vista puramente financeiro, ele ressalta que um *hypervisor* tipo 1 não exigiria o custo de um sistema operacional hospedeiro, embora, do ponto de vista prático, a discussão seja muito mais complexa e envolva todos os componentes e aspectos que compõem um cálculo do custo total de propriedade.

Alguns exemplos de soluções de *hypervisor* tipo 1 incluem *VMware vSphere*, *Microsoft Hyper-V* e *Citrix XenServer*. Dentre essas 3 soluções citadas como exemplo podemos destacar como o *Hyper-V* como sendo a única que é habilitada a partir de um sistema operacional já existente, neste caso o próprio *Windows* e que a partir de sua instalação permite a criação de sistemas convidados *Windows* ou *Linux* através de sua interface que pode ser observada Figura 3. (MICROSOFT, 2022)

**Figura 3** Hyper-V Manager

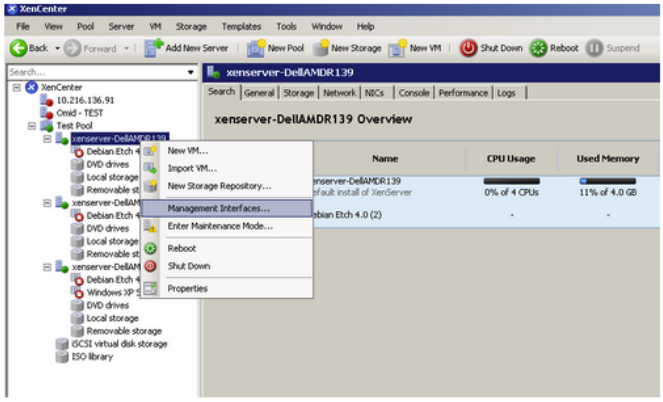



Fonte: MICROSOFT, 2022.

Nos dois outros exemplos uma vez que o *hypervisor* apenas um componente que viabiliza a criação de máquinas virtuais as configurações são acessadas por terminal ou através

da rede, para o *Citrix XenServer*, a interface de configuração (Figura 4) dever ser inicialmente habilitada pelo *console* e posteriormente acessado de outro computador e é possível criar máquinas *Windows* ou *Linux*. (CITRIX, 2023)

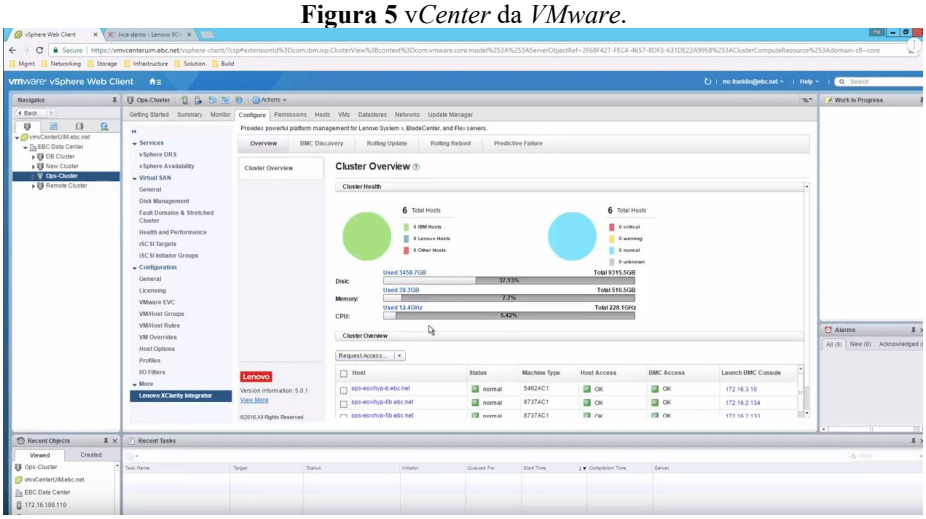
**Figura 4** Citrix XenServer Management Interface




Fonte: CITRIX, 2023.

Já no *VMware vSphere*, que utiliza o *vCenter* exibido na Figura 5 para sua configuração onde é possível a criação de máquinas com sistemas a opção de utilizar sistemas operacionais *Linux*, *Windows* ou *MAC OS*. Contudo, desde o final de 2022 a *VMware* está descontinuando o suporte e oferta de virtualização de sistemas operacionais convidados *MAC* (VMWARE, 2022)

**Figura 5** v*Center* da *VMware*.



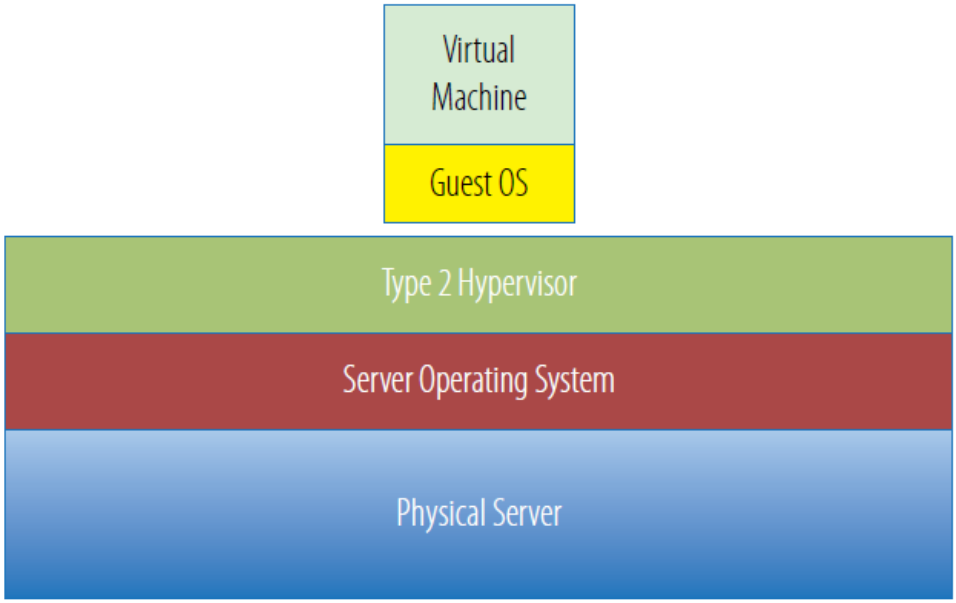

Fonte: VMWARE, 2023.

Dentre as 3 soluções citadas, a desenvolvida pela *VMware* é a única que permite virtualização de sistemas operacionais *Apple*

Já o *hypervisor* tipo 2, também conhecido como virtualização de nível de software, é a forma mais comum de virtualização e é baseada em software intermediário. Nesta forma de virtualização, o sistema operacional *host* é instalado em uma máquina física, e o sistema operacional convidado é instalado em uma máquina virtual dentro do sistema operacional *host*.

Portnoy (2016) explica que um *hypervisor* tipo 2 em si é um aplicativo que é executado em cima de um sistema operacional tradicional. As primeiras ofertas x86 eram do Tipo 2 porque esse era o caminho mais rápido para o mercado - o sistema operacional real já lidava com todos os recursos de hardware e o *hypervisor* aproveitaria essa capacidade.

**Figura 6** *hypervisor* tipo 2




Fonte: PORTNOY, 2016.

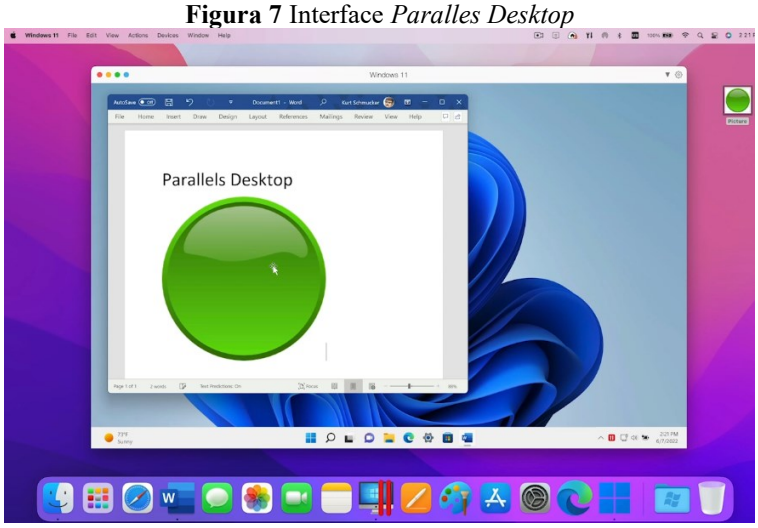
De acordo com Portnoy (2016), um benefício desse modelo é que ele pode suportar uma ampla variedade de hardwares uma vez que isso é herdado do sistema operacional que ele usa. Frequentemente, os *hypervisors* tipo 2 são fáceis de instalar e implantar, porque grande parte do trabalho de configuração de hardware, como rede e armazenamento, já foi feito pelo sistema operacional.

Portnoy (2016) ressalta que os *hypervisors* tipo 2 não são tão eficientes quanto os *hypervisors* tipo 1 devido a essa camada extra entre o *hypervisor* em si e o *hardware*. Sempre que uma máquina virtual realiza uma leitura de disco, uma operação de rede ou qualquer outra interação com o hardware, ela transfere essa solicitação para o *hypervisor*, assim como em um ambiente de *hypervisor* tipo 1. Ao contrário desse ambiente, o *hypervisor* tipo 2 precisa, em seguida, transferir a solicitação para o sistema operacional, que lida com as solicitações de *I/O*. O sistema operacional envia as informações de volta para o *hypervisor* e, em seguida, de volta para o convidado, adicionando duas etapas adicionais, tempo e recursos de processamento, a cada transação.

Portnoy (2016) destaca que os *hypervisors* tipo 2 também são menos confiáveis, pois apresentam mais pontos de falha. Qualquer coisa que afete a disponibilidade do sistema operacional subjacente também pode impactar o *hypervisor* e os convidados que ele suporta. Por exemplo, as atualizações de rotina do sistema operacional que exigem uma reinicialização do sistema também forçariam reinicializações de todas as máquinas virtuais nesse *host*.

Por fim, alguns exemplos de soluções de *hypervisor* tipo 2 incluem o *Oracle VirtualBox*, *VMware Workstation* e *Parallels Desktop*. Por serem caracterizadas como virtualizações por softwares, além e encontrarmos diferenças quanto aos sistemas operacionais convidados, ocorre também a diferença entre o sistema que as empresas oferecem as aplicações. O *Parallels Desktop* (Figura 7) é oferecido para plataformas *Mac* oferecendo virtualização de sistema *Windows* (PARALLES, 2023). Já os softwares *Virtualbox* (Figura 8) e *VMware Workstation* (Figura 9) permitem virtualizar sistemas *Windows*, *Mac*, *Linux* e *Solaris* e podem ser instalados em máquinas *Windows*, *Linux* ou *Mac.*

**Figura 7** Interface *Paralles Desktop*

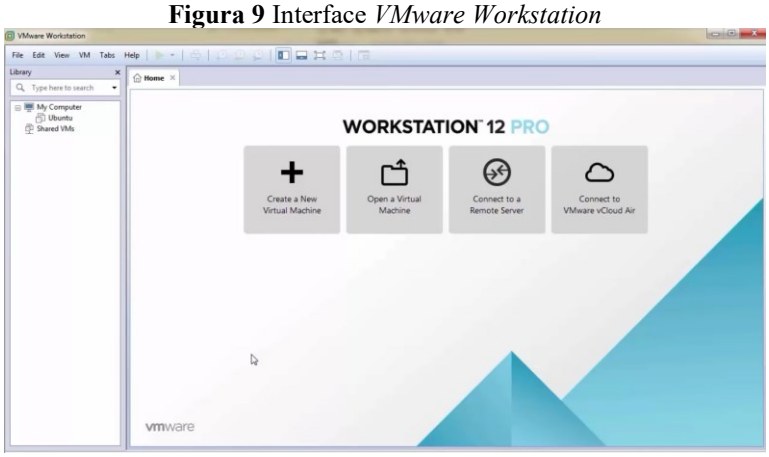



Fonte: PARALLES, 2023.

**Figura 8** Interface *Virtualbox*

Fonte: ORACLE, 2023.

**Figura 9** Interface *VMware Workstation*




Fonte: VMWARE, 2023.

## 5. Trabalhos Similares

O estudo realizado por Agard (2009), apresenta proposta similar a este trabalho explorando as características do software de virtualização *VMWare ESX Server* que utiliza um *hypervisor* de tipo 1. Após a apreensão teoria sobre a tema da virtualização e suas principais características, o autor segue para uma apresentação da ferramenta estudada na prática

realizando sua instalação em um conjunto de servidores de tamanho físico reduzido e utilizando uma ferramenta da fabricante *HP* para fazer o gerenciamento dos componentes físicos do sistema.

Apesar de serem da mesma fabricante o processo de configuração das máquinas ocorre através de softwares distintos, o *VMWare ESX Server* é o software responsável pelo gerenciamento dos sistemas operacionais convidados, mas também é necessário utilizar os softwares *VMWare Infrastructure Client, Infrastructure Update, vSphere Client e vSphere Host Update Utility* para configurar todo o sistema. (AGARD, 2009).

Em um trabalho apresentado por Arriel (2014) apresenta a comparação entre duas estratégias de virtualização, a virtualização tipo 1 utilizada pelo *VMware* e a tipo 2 utilizada pelo *Xen*. Os testes realizados pelo autor visaram a observação de desempenho processador, memória, disco, rede e sistema, onde de forma geral a virtualização completa apresentada pela *VMWare* presentou desempenho superior na maioria dos cenários.

Aguiar (2014), em sua tese de mestrado, aborda a questão da virtualização em sistemas embarcados multiprocessados. Seu objetivo principal é analisar e avaliar a utilização da virtualização nesse tipo de sistema, a fim de proporcionar uma solução mais eficiente e flexível em termos de gerenciamento de recursos. O estudo apresenta uma revisão de conceitos e tecnologias relacionadas à virtualização de sistemas embarcados, além de apresentar uma análise detalhada do desempenho do sistema virtualizado em relação ao sistema nativo. Os resultados indicam que a virtualização pode ser uma solução viável para sistemas embarcados multiprocessados, trazendo benefícios como maior flexibilidade e gerenciamento mais eficiente de recursos.

## 6. Metodologia

Para atingir os objetivos deste estudo de caso sobre o *WSL*, foram utilizados métodos e ferramentas que permitam uma análise detalhada e aprofundada das características e funcionalidades dessa tecnologia.

A metodologia deste trabalho está dividida nas etapas de revisão bibliográfica, onde foi feito um levantamento, para compor a fundamentação teórica, sobre os conceitos de virtualização e seus tipos, bem suas principais características. A etapa do estudo de caso, onde foi realizada uma pesquisa sobre o *WSL*, suas características, funcionamento e aplicabilidade, essa pesquisa foi realizada em buscando fontes confiáveis, incluindo artigos científicos, livros e sites oficiais da *Microsoft*.

Na sequência, foi feita a análise e discussão dos resultados obtidos através da revisão bibliográfica e do estudo de caso. Por fim, as conclusões, que foram obtidas através da comparação dos resultados das pesquisas realizadas, com os objetivos propostos. Além disso, foram identificadas as contribuições do *WSL* para a área de análise e desenvolvimento de sistemas.

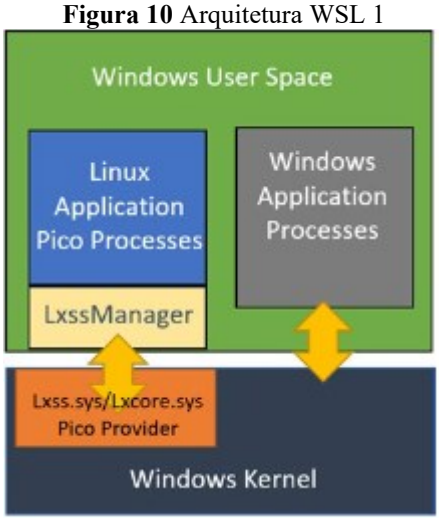
Com essas ferramentas e métodos, foi possível realizar uma análise do *WSL* e avaliar sua efetividade como solução de virtualização de sistemas operacionais no ambiente *Windows*.

## 7. Desenvolvimento

O *WSL* é uma camada de compatibilidade desenvolvida pela *Microsoft* lançada e em sua primeira versão, o *WSL* 1, em 2 de agosto de 2016 como parte integral de seu sistema operacional, o *Windows 10*. Nessa primeira versão, a arquitetura utilizada para a virtualização é a do tipo 2, onde como pode ser observado pela Figura 10 a máquina *Linux* é instalada dentro do espaço do usuário no *Windows 10* e tem como principal responsabilidade o encaminhamento de chamadas para a execução de binários *Linux* para o *Lxcore.sys* (responsável pela tradução

dos comandos para comandos NT) além de monitoramento das máquinas *Linux* e garantir a instalação e remoção de pacotes no sistema *Linux*. (BARNES, 2021).

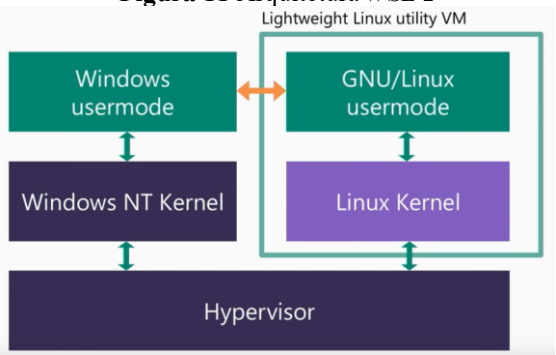
**Figura 10** Arquitetura WSL 1




Fonte: BARNES, 2021

Segundo Barnes (2021) neste tipo de virtualização o sistema operacional convidado não precisa ser instalado em sua totalidade uma vez que não há um kernel *Linux* nativo que fará o processamento dos comandos, este tipo de solução apresenta uma vantagem quando a eficiente energética é mais importante do que a performance. Já o *WSL 2,* lançado em 2019 possui uma arquitetura completamente diferente da implementada na primeira versão, trazendo o kernel do *Linux* e do próprio *Windows* em cima de um *HYPER-V*, caracterizando-o como uma virtualização do tipo 1 (Figura 11).

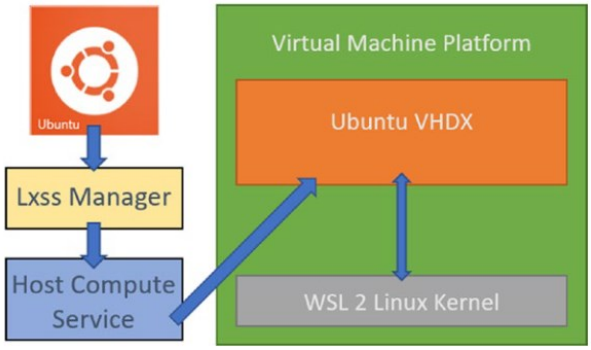
**Figura 11** Arquitetura *WSL 2*




Fonte: MICROSOFT, 2019

Desta forma processos executados passam a apresentar um fluxo conforme demonstrado na Figura 12, onde o *kernel Linux* garante o processamento nativo aos comandos e operações executados sem o risco de erros por ausência do um comando equivalente por parte do sistema hospedeiro conforme ocorre na virtualização tipo 1.

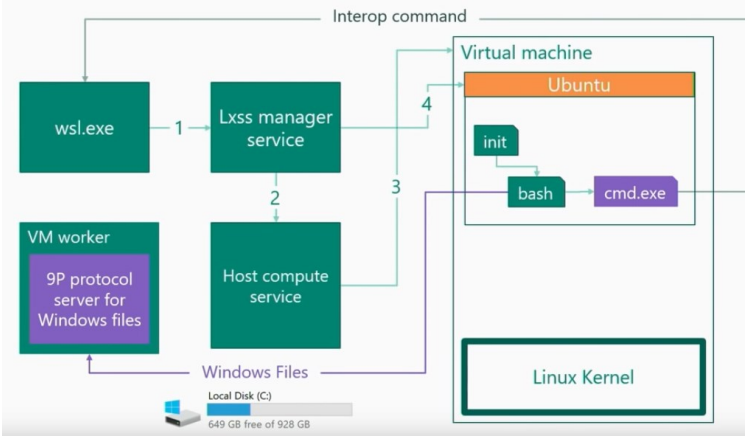
**Figura 12** Arquitetura do sistema operacional convidado




Fonte: BARNES, 2021

Nesta nova arquitetura o autor destaca alguns avanços e melhorias em relação ao *WSL* 1 como o ganho de performance pelo uso do próprio *kernel Linux* para a execução dos chamados do sistema, a virtualização de adaptadores de rede e de sistema de arquivos que permite que o sistema operacional *Windows* acesse arquivos do sistema *Linux* e vice e versa, como exemplificado pela imagem abaixo na apresentação de parte dos desenvolvedores da *Microsoft* em um evento em 2019.

**Figura 13** Protocolo de virtualização de sistema de arquivos




Fonte: MICROSOFT, 2019

Outra melhoria nesta segunda versão é a capacidade de executar aplicações *Linux* que possuem interface gráfica, algo não possível anteriormente e contribuiu a para a visão de sistema integrados não apenas em termos estruturais como também em usabilidade. Atualmente o *WSL* conta com a possibilidade de instalação das seguintes distribuições *Linux: Ubuntu, Kali Linux, Debian, OpenSUSE Tumbleweed* e *Server*, além de aceitar implementações de *kernel Linux* customizados.

## 8. Resultados e Discussões

Ao analisar a arquitetura do *WSL* e a proposta da *Microsoft* em construir uma ferramenta de virtualização de uso voltado para desenvolvedores, compreende-se a evolução de uma virtualização tipo 2 para a virtualização tipo 1 como um avanço importante não apenas para o possível ganho de performance como também para melhor refletir o comportamento de aplicações em ambientes *Linux* de produção uma vez que elas passam a ser executadas efetivamente pelo *kernel Linux*.

Apesar de sua implementação da virtualização de tipo 1 ter o *Windows* como um sistema "principal" uma vez que não é possível iniciar apenas o sistema *Linux*, ele oferece bom nível de integração entre os sistemas operacionais sem a necessidade de configurações por parte do usuário, permitindo com que por exemplo um projeto de aplicação web seja visualizado no editor *VS Code* pelo *kernel Windows,* mas a execução do projeto seja feita pelo *kernel Linux* e acessado pelo navegador no *Windows*.

Também foi possível observar o processo de instalação do *WSL* e das respectivas distribuições *Linux* como razoavelmente de baixa complexidade, permitindo que desenvolvedores pouco experientes tentem aderir a solução. Desde que o usuário esteja em um sistema *Windows* compatível não é necessária uma instalação limpa do sistema operacional, o recurso é em grande parte habilidade por interface gráfica conforme janela exibida a direita na Figura 14 e as distribuições disponível para instalar através do *Microsoft Store* na esquerda.

**Figura 14** Distribuições Linux e módulo WSL

Fonte: Próprio autor

Outra característica valiosa nas soluções de virtualização é a capacidade dos sistemas operacionais convidados serem facilmente replicados e isto também está presente na ferramenta da *Microsoft* através dos comandos *export* e *import* que permite ao usuário facilmente exportar e replicar o ambiente localmente ou em outro equipamento.

Por fim, no tange a disponibilidade de recursos e outras integrações foi observado que empresas como *NVIDIA* e *Docker* estão incorporando o *WSL* em seus produtos visando melhorar a experiência e satisfação dos usuários. A *NVIDIA*, por exemplo, anunciou recentemente a disponibilidade do *CUDA* (uma plataforma de computação paralela) no *WSL 2*. Isso permite que desenvolvedores usem a plataforma *NVIDIA* em seus projetos diretamente no *Windows 10*, sem a necessidade de usar uma máquina virtual separada ou alternar para outro sistema operacional.

Já o *Docker*, uma plataforma de desenvolvimento de aplicativos, lançou o *Docker* Desktop para *Windows* com suporte para *WSL 2*. Com essa integração, é possível executar contêineres *Docker* diretamente no *WSL*, permitindo que desenvolvedores trabalhem com um ambiente de desenvolvimento mais integrado e completo.

Este aumento nas funcionalidades e na adesão do *WSL* por outras ferramentas de desenvolvimento pode contribuir de forma positiva uma vez que viabiliza uma gama maior de casos de uso compatíveis com a solução.

## 9. Conclusões

Com base no estudo de caso realizado sobre o *WSL*, relacionando suas características com os conceitos teóricos de virtualização de sistemas operacionais e identificando suas diferenças em relação a outras soluções, é possível chegar a algumas conclusões relevantes.

Diante dos resultados obtidos, podemos concluir que o *WSL* é uma solução efetiva e promissora para a virtualização de sistemas operacionais *Linux* no ambiente *Windows*. Sua capacidade de integrar as funcionalidades do *Linux* ao *Windows*, de forma rápida e sem a configurações complexas, torna-o uma ferramenta valiosa para desenvolvedores. Ao permitir a execução de sistemas operacionais *Linux* de forma integrada, o WSL oferece flexibilidade, produtividade e eficiência no processo de desenvolvimento de software. Alguns dos possíveis benefícios encontrados em seu uso, foram a execução de ferramentas do *Linux* em sistemas *Windows* e a facilidade na configuração de ambientes de desenvolvimento, o que pode ser um fator que contribui para sua crescente popularidade no mercado de trabalho. Com base nesses resultados, podemos afirmar que o *WSL* é uma tecnologia que merece atenção e consideração por parte dos profissionais de TI e das empresas do setor, e a tendência é que a sua importância continue crescendo no futuro.

Embora este não seja um objetivo do presente trabalho, durante o levantamento bibliográfico não foi possível encontrar registros de grandes corporações que declararam abertamente utilizar o *WSL* em seu ambiente de desenvolvimento, apenas de forma confidencial foi levantada uma empresa da cidade de São José do Rio Preto que por relato de um de seus funcionários parece já adotar a solução para seu ambiente de desenvolvimento. Os resultados, contudo, revelaram que a demanda por profissionais com conhecimento em *WSL* está aumentando no mercado de trabalho, tornando-se um requisito comum em vagas na área de tecnologia. Isso ressalta a importância de dominar essa ferramenta, que pode ser um diferencial para os desenvolvedores que desejam se destacar.

No entanto, é importante reconhecer as limitações deste estudo. Embora tenhamos realizado uma pesquisa aprofundada e utilizado fontes confiáveis, este trabalho não abrange todas as especificidades e pormenores do uso do *WSL*, o que significa que algumas características importantes podem não ter sido abordadas.

Considerando as limitações identificadas, recomendamos que futuras pesquisas sejam realizadas para aprofundar ainda mais o estudo do *WSL* e sua aplicabilidade em diferentes contextos. Alguns tópicos que podem ser abordados por pesquisas futuras, são o de casos específicos de grandes corporações que utilizam o *WSL,* seu impacto e benefícios em ambientes de desenvolvimento, estudos comparativos entre o *WSL* e outras soluções de virtualização, a segurança do *WSL* e a evolução do *WSL* ao longo do tempo.

## 10. Referências

AGARD, Helen Danielsen Pena. **VIRTUALIZAÇÃO**: estudo das principais características e ferramentas com vmware esx server. 2009. 54 f. Monografia (Especialização) - Curso de Engenharia de Computação, Universidade São Francisco, Campinas, 2009.

AGUIAR, Alexandra da Costa Pinto de. **On the virtualization of multiprocessed embedded systems**. 2013. 109 f. Tese (Doutorado em Ciência da Computação) - Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2013. Disponível em: https://tede2.pucrs.br/tede2/handle/tede/5252

ARRIEL, Jhonny *et al*. **Análise comparativa entre as ferramentas de virtualização VMware e Xen**. 2014. 51 f. TCC (Graduação) - Curso de Sistemas de Informação, Ciência da Computação, Universidade Federal de Lavras, Lavras, 2014. Disponível em: http://repositorio.ufla.br/bitstream/1/30696/1/TCC_An%C3%A1lise_comparativa_entre_as_ferramentas_de_virtualiza%C3%A7%C3%A3o_vmware_e_xen.pdf. Acesso em: 2 fev. 2023.

BARNES, Hayden *et al*. **Pro Windows Subsystem for Linux (WSL)**: powerful tools and practices for cross-platform development and collaboration. Columbus: Apress, 2021.

CARISSIMI, Alexandre *et al*. Virtualização: da teoria a soluções. **XXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, Rio de Janeiro, v. 1, n. 1, p. 173-206, mar. 2008.

CITRIX. Citrix, 2023. All in one Workspace Solution for Secure Access to Apps and Data. Disponível em: https://support.citrix.com. Acessado em: 1 abril 2023

DOCKER Desktop WSL 2 backend on Windows. Disponível em: https://docs.docker.com/desktop/windows/wsl/. Acesso em: 12 abr. 2023.

MICROSOFT (org.). **Introduction to Hyper-V on Windows 10**. 2022. Disponível em: https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/. Acesso em: 21 mar. 2023.

MICROSOFT (org.). **The new Windows subsystem for Linux architecture:** a deep dive - BRK3068. 2019. Disponível em: https://youtu.be/lwhMThePdIo?list=LL. Acesso em: 21 mar. 2023.

MICROSOFT. **What is the Windows Subsystem for Linux?**. Microsoft, 2022. Disponível em: https://learn.microsoft.com/en-us/windows/wsl/about. Acessado em: 15 fev. 2023

NVIDIA. **Announcing Cuda On Windows Subsystem For Linux 2**. Disponível em: https://developer.nvidia.com/blog/announcing-cuda-on-windows-subsystem-for-linux-2/. Acesso em: 10 abr. 2023.

ORACLE, Virtualbox, 2023. Disponível em: https://www.virtualbox.org/. Acesso em: 1 abril 2023.

PARALLELS, Parallels, 2023. Virtualização MAc e Windows, Remote Application Server, Soluções de gerenciamento Mac. Disponível em: https://www.parallels.com/br/. Acesso em: 1 abril 2023.

PORTNOY, M. (2016). **Virtualization Essentials**, Second Edition. John Wiley & Sons, Inc.

RAJ, Tara. **What's new in WSL in Windows 10 Fall Creators Update**. Microsoft, 2017 Disponível em: https://devblogs.microsoft.com/commandline/whats-new-in-wsl-in-windows-10-fall-creators-update/. Acessado em: 10 mar. 2023

RANDAL, Allison *et al*. The Ideal Versus the Real: revisiting the history of virtual. **Association For Computing Machinery.** New York, jan. 2021. Seção 5, p. 1-31.

VAGA de Desenvolvedor .Net Sr. - Regional São Paulo. Disponível em: https://www.catho.com.br/vagas/desenvolvedor-.net-sr./22449441/. Acesso em: 25 abr. 2023.

VMWARE, Vmware, 2023. Delivering a digital foundation for business. Disponível em: https://www.vmware.com/. Acesso em: 1 abril 2023.