**Guia rápido do WSL2 + Docker**

**O que é o WSL2**

Em 2016, a Microsoft anunciou a possibilidade de rodar o Linux dentro do Windows 10 como um subsistema e o nome a isto foi dado de **WSL** ou **Windows Subsystem for Linux**.

O acesso ao sistema de arquivos no Windows 10 pelo Linux era simples e rápido, porém não tinhamos uma execução completa do kernel do Linux, além de outros artefatos nativos e isto impossibilitava a execução de várias tarefas no Linux, uma delas é o Docker.

Em 2019, a Microsoft anunciou o **WSL 2**, com uma dinâmica aprimorada em relação a 1ª versão:

* Execução do kernel completo do Linux.
* Melhor desempenho para acesso aos arquivos dentro do Linux.
* Compatibilidade completa de chamada do sistema.

O WSL 2 já estava disponível na versão **Insider** do Windows 10, mas na última semana de maio de 2020 passou a estar disponível em final release na atualização **20.04** do Windows 10.

**Atualização** A partir de 21 de agosto de 2020, o WSL 2 também está disponível nas edições 1903 e 1909, porém somente em sistemas x64.

Com WSL 2 é possível executar Docker no Linux usando o Windows 10.

Compare as versões: <https://docs.microsoft.com/pt-br/windows/wsl/compare-versions>

**O que é Docker**

Docker é uma plataforma open source que possibilita o empacotamento de uma aplicação dentro de um container. Uma aplicação consegue se adequar e rodar em qualquer máquina que tenha essa tecnologia instalada.

**Porque usar WSL 2 + Docker para desenvolvimento**

Configurar ambientes de desenvolvimento no Windows sempre foi burocrático e complexo, além do desempenho de algumas ferramentas não serem totalmente satisfatórias.

Com o nascimento do Docker este cenário melhorou bastante, pois podemos montar nosso ambiente de desenvolvimento baseado em Unix, de forma independente e rápida, e ainda unificada com outros sistemas operacionais.

Veja nossa **live sobre WSL 2 + Docker no canal Full Cycle**: <https://www.youtube.com/watch?v=g4HKttouVxA>.

Para se utilizar Docker no Windows temos três versões:

* Docker Toolbox.
* Docker Desktop com Hyper-V.
* Docker Desktop com WSL2.

**Docker Toolbox**

Roda em cima do programa de virtualização de sistemas da Oracle, chamado de **VirtualBox**. O desempenho do Docker Toolbox para muitas aplicações/ferramentas pode ser muito ruim, inviabilizando seu uso.

**Docker Desktop com Hyper-V**

Roda em cima do **Hyper-V** da Microsoft em vez de usar o VirtualBox usando pelo Docker Toolbox. O Docker Desktop com Hyper-V necessita da versão **PRO** do Windows 10, portanto é necessário compra-la se você não a tem.

O Hyper-V costuma requerer muitos recursos da máquina e apesar do desempenho ser melhor que o Docker Toolbox, a máquina pode ficar lenta para se utilizar outras coisas no Windows.

*A Docker já anunciou que vai remover o suporte ao Hyper-V futuramente.*

**Docker Desktop com WSL2**

Roda em cima do **Virtual Machine Platform** em vez de usar o VirtualBox ou Hyper-V. Se integra com o WSL2 permitindo rodar o Docker dentro do ambiente do Linux. Não é necessário adquirir licença PRO do Windows 10, tem um grande desempenho e consome menos recursos quando comparado ao Docker Toolbox ou Docker Desktop com Hyper-V.

Temos a grande vantagem de se trabalhar totalmente dentro do Linux para desenvolvimento, portanto, usar WSL2 + Docker é a melhor maneira de se desenvolver aplicações no Windows.

**Requisitos mínimos**

* Windows 10 Home ou Professional com versão **20.04** ou superior.

**Atualização** A partir de 21 de agosto de 2020, o WSL 2 também está disponível nas edições 1903 e 1909, porém somente em sistemas x64.

* Uma máquina compatível com virtualização (verifique a disponibilidade de acordo com a marca do seu processador. Se sua máquina for mais antiga pode ser necessária habilita-la na BIOS).
* Pelo menos 4GB de memória RAM.

**Instalação do WSL 2**

**Instale a versão 20.04 do Windows 10**

O WSL 2 só funciona no Windows 10 **20.04** ou superior.

Pode ser que seu Windows 10 já seja igual ou superior a 20.04, verifique isto acessando o menu de notificações perto do relógio > Todas as configurações > Sistema > Sobre.

[](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/img/windows_10_2004.png)

Se seu Windows 10 não for 20.04 baixe o pacote de atualização <https://go.microsoft.com/fwlink/?LinkID=799445>. A instalação pode demorar aproximadamente **1 hora**.

**Habilite o WSL no Windows 10**

Execute os seguintes comandos no PowerShell em modo administrador:

dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

Abra o PowerShell e digite o comando wsl, se não funcionar reinicie sua máquina.

**Instale o WSL 2 no Windows 10**

Baixe o Kernel do WSL 2 neste link: <https://docs.microsoft.com/pt-br/windows/wsl/wsl2-kernel> e instale o pacote.

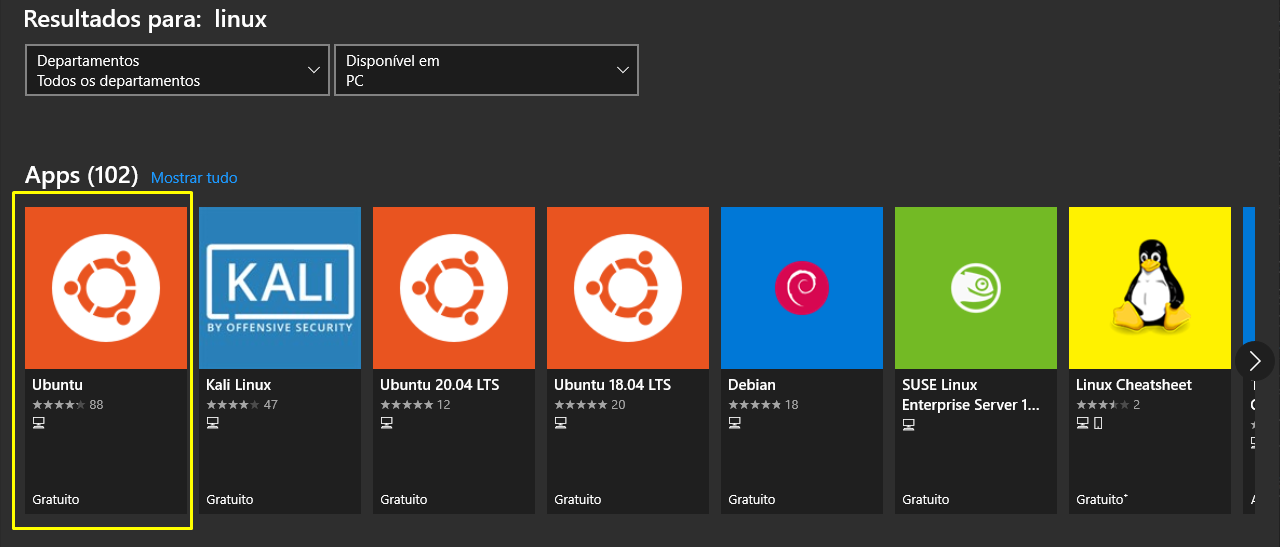
**Atribua a versão default do WSL para a versão 2**

A versão 1 do WSL é a padrão no momento, atribua a versão default para a versão 2, assim todas as distribuições Linux instaladas serão já por default da versão 2. Execute o comando com o PowerShell:

wsl --set-default-version 2

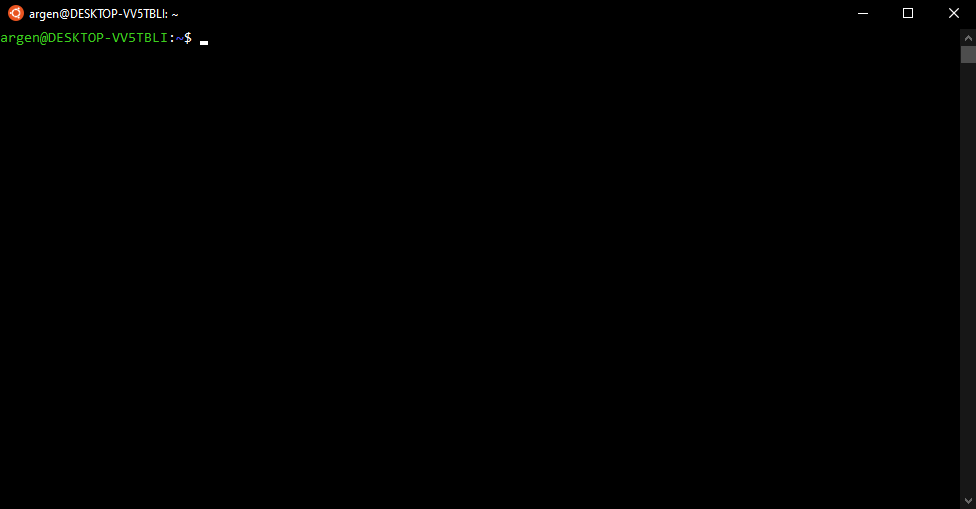
**Escolha sua distribuição Linux no Windows Store**

Escolha sua distribuição Linux preferida no aplicativo Windows Store, sugerimos o Ubuntu por ser uma distribuição popular e que já vem com várias ferramentas instaladas por padrão.

[](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/img/distribuicoes_linux.png)

Ao iniciar o Linux instalado, você deverá criar um **nome de usuário** que poderá ser o mesmo da sua máquina e uma **senha**, este será o usuário **root da sua instância WSL**.

Parabéns, seu WSL2 já está funcionando:

[](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/img/wsl2_funcionando.png)

**(Opcional) Desinstale o Hyper-V**

Agora que temos o WSL 2 não precisamos mais do Hyper-V, desabilite-o em Painel de Controle > Programas e Recursos (se você tiver instalado o Hyper-V).

**(Opcional) Alterar a versão do WSL 1 de uma distribuição para a versão 2**

Se você já tiver o WSL 1 na máquina e acabou de instalar a versão 2, pode-se converter sua distribuição Linux WSL 1 para WSL 2, execute o comando com o PowerShell:

wsl --set-version <distribution name> 2

Isto pode demorar muitos minutos.

**(Opcional) Usar Windows Terminal como terminal padrão de desenvolvimento para Windows**

Uma deficiência que o Windows sempre teve era prover um terminal adequado para desenvolvimento. Agora temos o **Windows Terminal** construído pela própria Microsoft que permite rodar terminais em abas, alterar cores e temas, configurar atalhos e muito mais.

Instale-o pelo Windows Store e use estas [configurações padrões](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/windows-terminal-settings.json) para habilitar WSL 2, Git Bash e o tema drácula e alguns atalhos.

Para sobrescrever as configurações **clique a seta para baixo do lado das abas e em configurações**, abrirá as configurações do Windows Terminal, apenas cole o conteúdo do arquivo JSON e salve.

**O que o WSL 2 pode usar de recursos da sua máquina**

Podemos dizer que o WSL 2 tem acesso quase que total ao recursos de sua máquina. Ele tem acesso por padrão:

* A todo disco rígido.
* A usar completamente os recursos de processamento.
* A usar 80% da memória RAM disponível.
* A usar 25% da memória disponível para SWAP.

Isto pode não ser interessante, uma vez que o WSL 2 pode usar praticamente todos os recursos de sua máquina, mas podemos configurar limites.

Crie um arquivo chamado .wslconfig na raiz da sua pasta de usuário (C:\Users\<seu\_usuario>) e defina estas configurações:

[wsl2]

options=metadata,umask=22,fmask=11

memory=8GB

processors=4

swap=2GB

Estes são limites de exemplo e as configurações mais básicas a serem utilizadas, configure-os às suas disponibilidades. Para mais detalhes veja esta documentação da Microsoft: <https://docs.microsoft.com/pt-br/windows/wsl/wsl-config#wsl-2-settings>.

Para aplicar estas configurações é necessário reiniciar as distribuições Linux, então sugerimos executar no PowerShell o comando: wsl --shutdown (Este comando vai desligar todas as instâncias WSL 2 ativas e basta abrir o terminal novamente para usa-la já com as novas configurações).

**Integrar Docker com WSL 2**

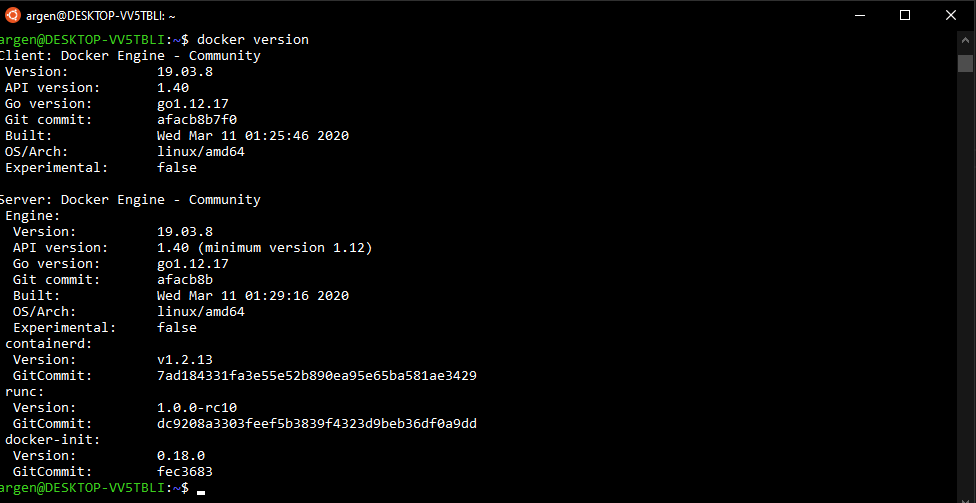
**Instale o Docker Desktop**

Baixe neste link: <https://hub.docker.com/editions/community/docker-ce-desktop-windows> e instale o Docker Desktop.

**Habilite o Docker dentro do WSL 2**

Clique no ícone do Docker perto do relógio -> Settings -> Settings no topo -> Resources -> WSL Integration.

Habilite Enable integration with my default WSL distro e habilite sua versão Linux.

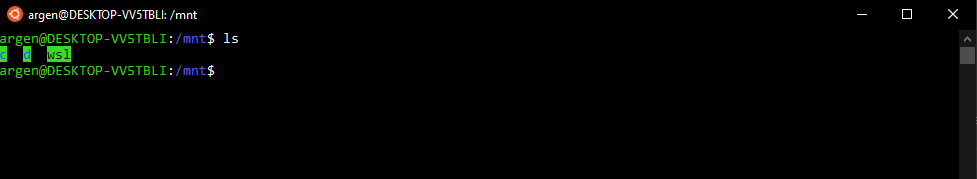
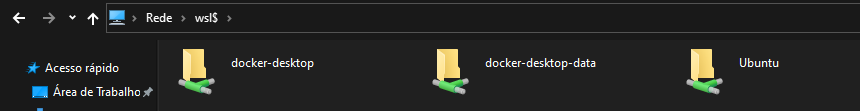
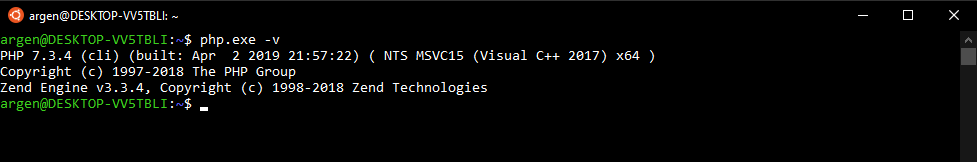
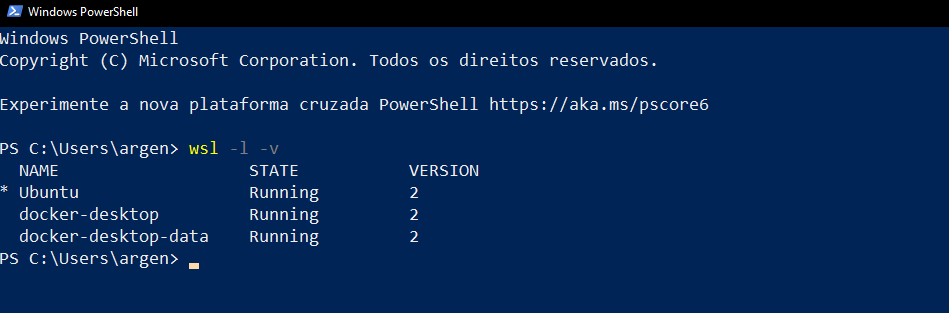
[](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/img/docker_funcionando_dentro_do_wsl2.png)

Parabéns, você agora tem um excelente ambiente de desenvolvimento com WSL 2 e Docker.

**Use BuildKit and multi-stage builds**

Acrescente export DOCKER\_BUILDKIT=1 no final do arquivo .profile do seu usuário do Linux para ganhar mais performance ao realizar builds com Docker. Execute o comando source ~/.profile para carregar esta variável de ambiente no ambiente do seu WSL 2.

**Dicas e truques básicos com WSL 2**

* A performance do WSL 2 está em se executar tudo dentro do Linux, por isso evite executar seus projetos com ou sem Docker do caminho /mnt/c, pois você perderá performance.
* Para abrir o terminal do WSL basta digitar o nome da distribuição no menu Iniciar ou executar C:\Windows\System32\wsl.exe.
* O sistema de arquivos do Windows 10 é acessível em /mnt. [](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/img/mount_no_wsl2.png)
* É possível acessar o sistema de arquivos do Linux pela rede do Windows, digite \\wsl$ no Windows Explorer. [](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/img/acessando_wsl2_no_explorer.png)
* É possível acessar uma pasta no Windows Explorer digitando o comando explorer.exe ..
* É possível abrir uma pasta ou arquivo com o Visual Studio Code digitando o comando code . ou code meu\_arquivo.ext.
* Incrivelmente é possível acessar executáveis do Windows no terminal do Linux executando-os com .exe no final (não significa que funcionarão corretamente). [](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/img/executaveis_do_windows_no_wsl2.png)
* É possível executar algumas aplicações gráficas do Linux com WSL 2. Leia este tutorial: <https://medium.com/@dianaarnos/aplica%C3%A7%C3%B5es-gr%C3%A1ficas-no-wsl2-e0a481e9768c>.
* Execute o comando wsl -l -v com o PowerShell para ver as versões de Linux instaladas e seu status atual(parado ou rodando). [](https://github.com/codeedu/wsl2-docker-quickstart/blob/master/img/verificando_distribuicoes_instaladas_do_linux_no_wsl2.png)
* Execute o comando wsl --shutdown com o PowerShell para desligar todas as distribuições Linux que estão rodando no mento (ao executar o comando, as distribuições do Docker também serão desligadas e o Docker Desktop mostrará uma notificação ao lado do relógio perguntando se você quer iniciar as distribuições dele novamente, se você não aceitar terá que iniciar o Docker novamente com o ícone perto do relógio do Windows).
* Execute com o PowerShell o comando wsl --t <distribution name> para desligar somente uma distribuição Linux específica.
* Se verificar que o WSL 2 está consumindo muitos recursos da máquina, execute os seguintes comandos dentro do terminal WSL 2 para liberar memória RAM:

echo 1 | sudo tee /proc/sys/vm/drop\_caches

echo 3 | sudo tee /proc/sys/vm/drop\_caches

**Dúvidas**

* O WSL 2 funciona junto com outras máquinas virtuais como **VirtualBox** ou **VMWare**? Siga a [referência](https://docs.microsoft.com/pt-br/windows/wsl/wsl2-faq#will-i-be-able-to-run-wsl-2-and-other-3rd-party-virtualization-tools-such-as-vmware-or-virtualbox)