



Andrei equipe Dia 2 4 de 6

74 Windows

1. Introdução

Para configurar o ambiente de desenvolvimento no Windows, precisamos ter instaladas as duas ferramentas principais para rodar o nosso projeto clone do TabNews, que são o Node.js e o Docker. Neste tutorial nós iremos detalhar como fazer isso, além de sugerir algumas outras ferramentas adicionais para otimizar o trabalho.

● **Atenção!** Essa é uma aula com conteúdo de nível avançado. Caso você seja iniciante, recomendamos que siga com o desenvolvimento do projeto no ambiente do [Codespaces](#). Mais pra frente, quando estiver mais confortável, você pode voltar aqui e configurar o seu ambiente local, sem prejuízo algum.

2. Docker no Windows

O Docker é um recurso fantástico e nós teremos a oportunidade de nos aprofundarmos mais nele ao chegarmos no Dia 17, quando formos configurar o nosso banco de dados local. Mas, por enquanto, basta dizer que ele nos permite executar programas em ambientes isolados e padronizados, conhecidos como *contêineres*, com a garantia de que esses programas funcionarão da exata mesma maneira em qualquer máquina, independentemente das configuração de hardware ou software desta.

No Windows, existem três maneiras de se instalar e usar o Docker:

1. Docker Desktop com a base no Hyper-V;
2. Docker Desktop com a base no WSL;
3. Docker Engine diretamente dentro do WSL;

Mas qual a diferença entre elas, e qual devo escolher? A fim de tomar uma decisão bem informada vamos primeiro esclarecer brevemente alguns termos para ao final chegar à solução mais adequada ao nosso contexto. Vamos começar pelo **Hyper-V** e **WSL**, que dizem respeito à camada base (ou *backend*) que oferece o suporte para o Docker, e em seguida explicar os termos **Docker Engine** e **Docker Desktop** que dizem respeito à aplicação em si, e à forma de interação com o usuário.

2.1 Esclarecendo o que é Hyper-V e WSL: o *backend* do Docker no Windows

A "casa" do Docker é o Linux. É onde foi projetado originalmente, e onde também estão os recursos nativos dos quais depende, como nós veremos mais na frente no curso. Portanto, para que ele possa funcionar no Windows, é preciso que haja "por baixo" dele uma camada adicional que simule o seu ambiente original. E é aí que entram o **Hyper-V** e o **WSL**, as "pontes" entre o Docker e o Windows.

De modo bem simples, tanto o **Hyper-V** como o **WSL** são programas que utilizam a tecnologia de máquinas virtuais (as famosas **VM's**). Essa tecnologia permite que você simule, dentro do seu sistema operacional, um computador virtual com seus recursos de hardware e sistema operacional próprios. A diferença é que, enquanto o **Hyper-V** é uma plataforma de virtualização completa, que permite a criação de várias máquinas virtuais com diferentes sistemas operacionais (Windows ou Linux), incluindo o gerenciamento manual dos recursos de hardware dessas máquinas, o **WSL** consiste em apenas uma máquina virtual que roda o kernel do Linux, e cujo gerenciamento dos recursos de hardware é feito automaticamente e de modo abstruído do usuário.

Ok, mas qual devo escolher? Bem, como o nosso objetivo é rodar apenas o Docker, o **Hyper-V** é muito mais do que nós precisamos, e, além de adicionar uma complexidade a mais no processo, está disponível [apenas nas versões PRO do Windows](#) 10 ou 11. Portanto, o **WSL** é a solução mais apropriada, pois nos oferece uma máquina virtual pré-configurada, leve, e integrada ao Windows.

2.2 Esclarecendo o que é Docker Engine e Docker Desktop

Agora que já definimos como nós iremos fornecer o ambiente Linux para o Docker no Windows, vamos entender quais são as outras duas formas de se instalar e utilizar o Docker.

Novamente, de modo bem simples, o **Docker Engine** é o núcleo da ferramenta e o responsável pelo trabalho de criação, execução, e gerenciamento dos *contêineres*. É o motor que faz as coisas se mexerem. Por isso o nome "*engine*". Já o **Docker Desktop** é uma solução mais completa, que inclui o **Docker Engine**, mas oferece também uma interface gráfica para facilitar as configurações. Aproveitando a analogia, seria como a lataria do carro, com vários acessórios opcionais.

Por conta dos recursos adicionais, naturalmente o **Docker Desktop** é mais pesado e consome mais recursos da máquina hospedeira^[1]. Por esse motivo, e porque a nossa interação com o Docker durante as aulas será exclusivamente via linha de comando no terminal do Linux^[2], o caminho recomendado será instalar o **Docker Engine** diretamente dentro do **WSL**.

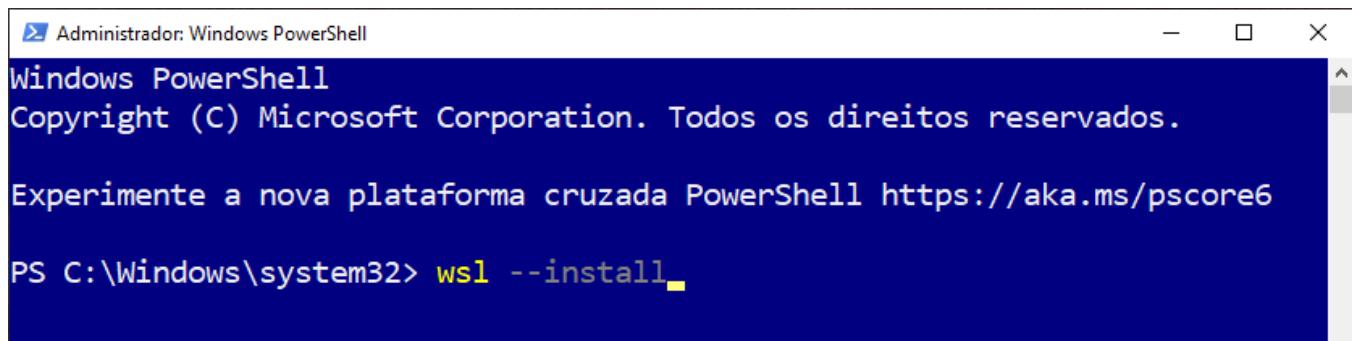
2.3 Instalando o WSL 2

Agora com o nosso caminho de instalação definido, vamos colocar a mão na massa de fato, primeiramente instalando o **WSL 2**.

Os passos descritos abaixo são baseados na documentação oficial da Microsoft, sendo necessária a execução de apenas um comando para desencadear o processo automático. O requisito para a instalação é ter o Windows 11, ou o Windows 10 na versão 20H1 e superior (Build 19041 e superior)^[3].

Passo 1: Habilitar recursos e instalar Ubuntu

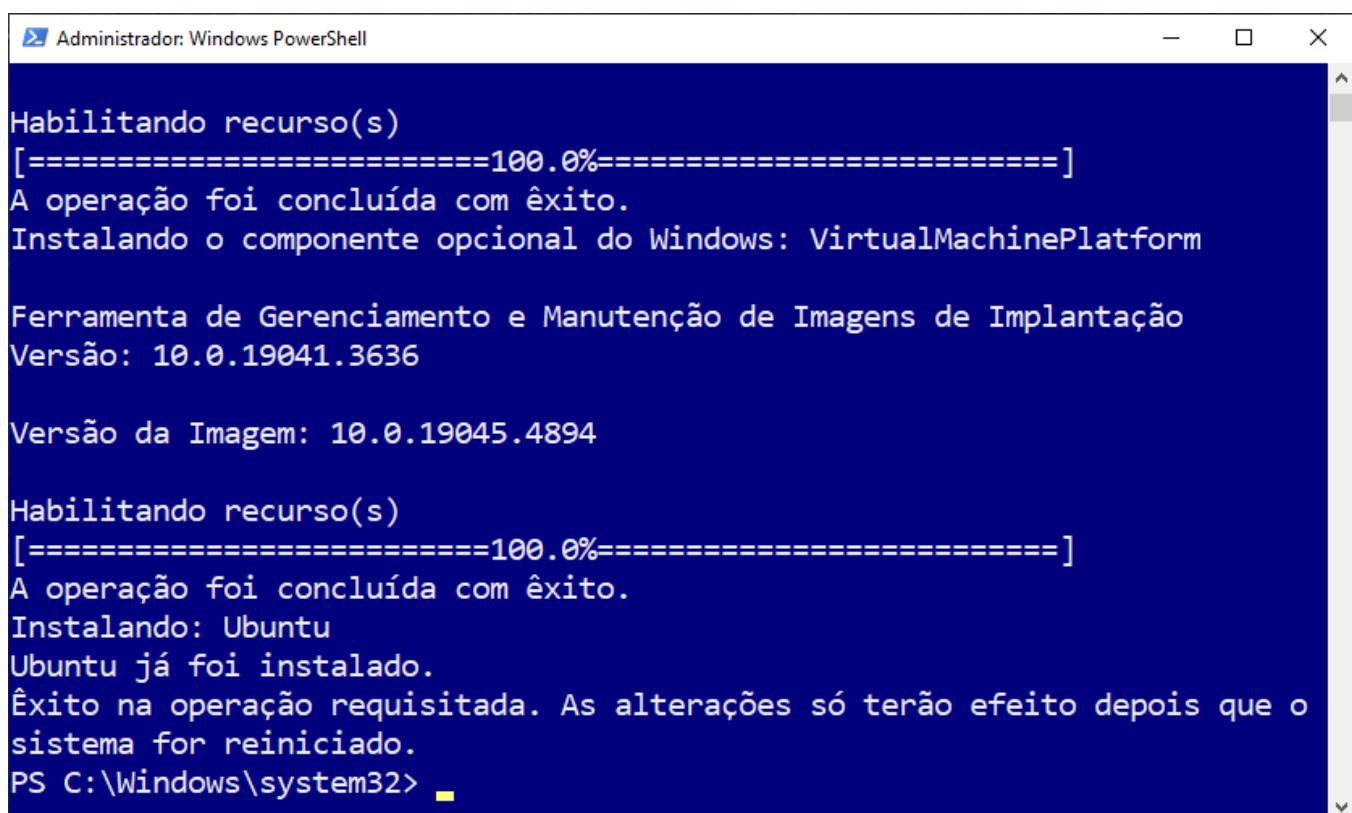
Com o PowerShell (ou o Prompt de Comando do Windows) aberto no modo de administrador, executar o comando `wsl --install`. Esse comando habilitará os recursos necessários para executar o WSL e irá instalar a distribuição Ubuntu do Linux. Para que a instalação seja concluída será necessário reiniciar o computador.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> wsl --install
```



```
Habilitando recurso(s)
[=====100.0%=====]
A operação foi concluída com êxito.
Instalando o componente opcional do Windows: VirtualMachinePlatform

Ferramenta de Gerenciamento e Manutenção de Imagens de Implantação
Versão: 10.0.19041.3636

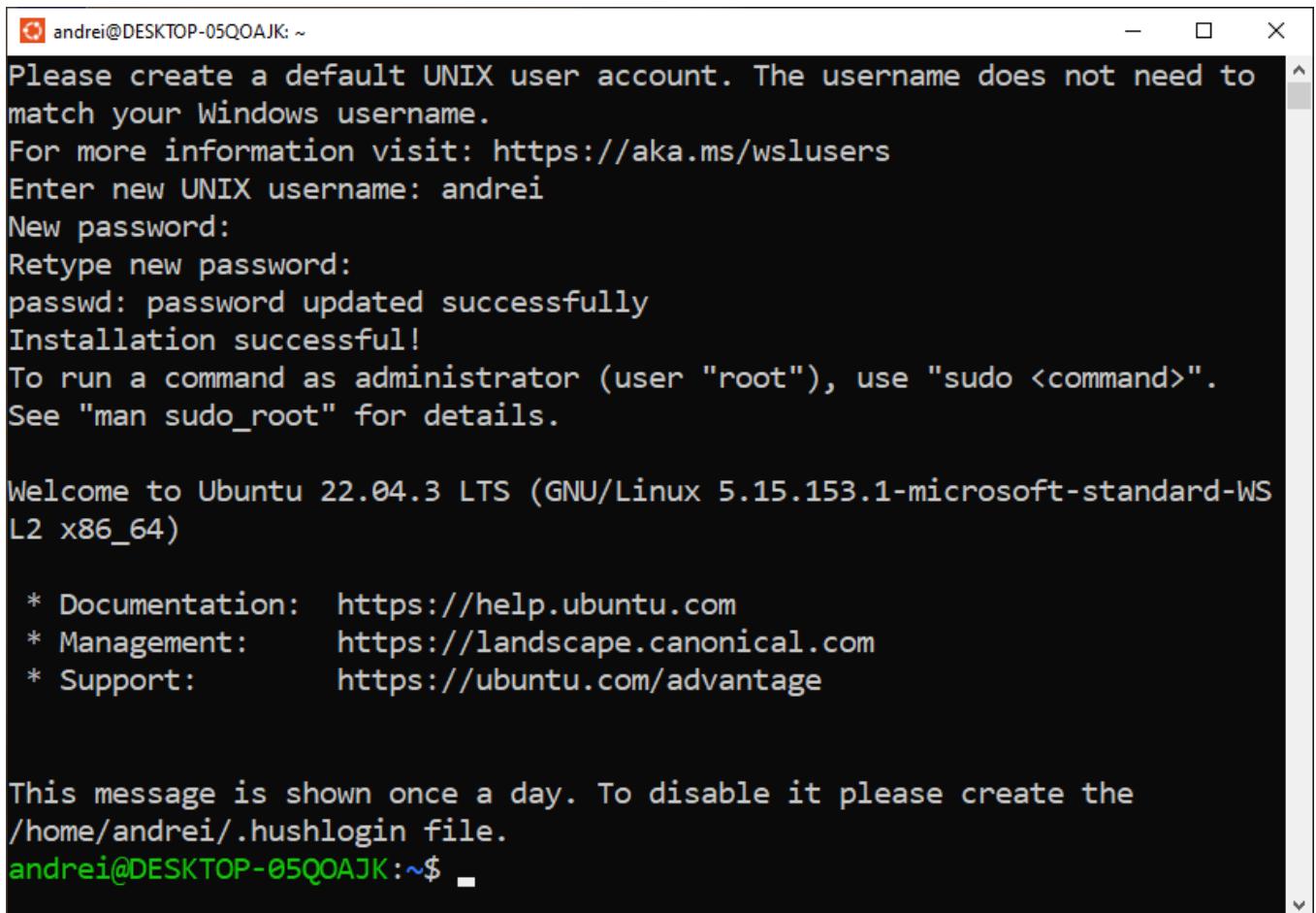
Versão da Imagem: 10.0.19045.4894

Habilitando recurso(s)
[=====100.0%=====]
A operação foi concluída com êxito.
Instalando: Ubuntu
Ubuntu já foi instalado.
Êxito na operação requisitada. As alterações só terão efeito depois que o sistema for reiniciado.
PS C:\Windows\system32>
```

Passo 2: Criar usuário no Ubuntu

Após reiniciar o computador, caso uma janela de terminal do Ubuntu não abra automaticamente, vá no menu iniciar e busque por "Ubuntu". Clicando no ícone será aberto o terminal da distribuição. Após aberto, a distribuição será inicializada, solicitando a criação de

um usuário e senha. *Obs.:* ao digitar a senha, nada será exibido. Isso é um recurso de segurança padrão do terminal.



```
andre@DESKTOP-05QOAJK: ~
Please create a default UNIX user account. The username does not need to
match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: andrei
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WS
L2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/andrei/.hushlogin file.
andre@DESKTOP-05QOAJK:~$
```

2.4 Instalando o Windows Terminal

Antes das próximas etapas é recomendável a instalação do Windows Terminal, que é uma interface moderna, unificada, e customizável para vários shells. Ele irá facilitar a nossa vida na hora de colar e executar os comandos de múltiplas linhas durante o processo de instalação do Docker. No Windows 11 ele já está presente nativamente, mas no Windows 10 é preciso instalá-lo via Microsoft Store.



Store



Entrar



Início

Aplicativos

Jogos

Entretenimento

Sobre

Windows Terminal

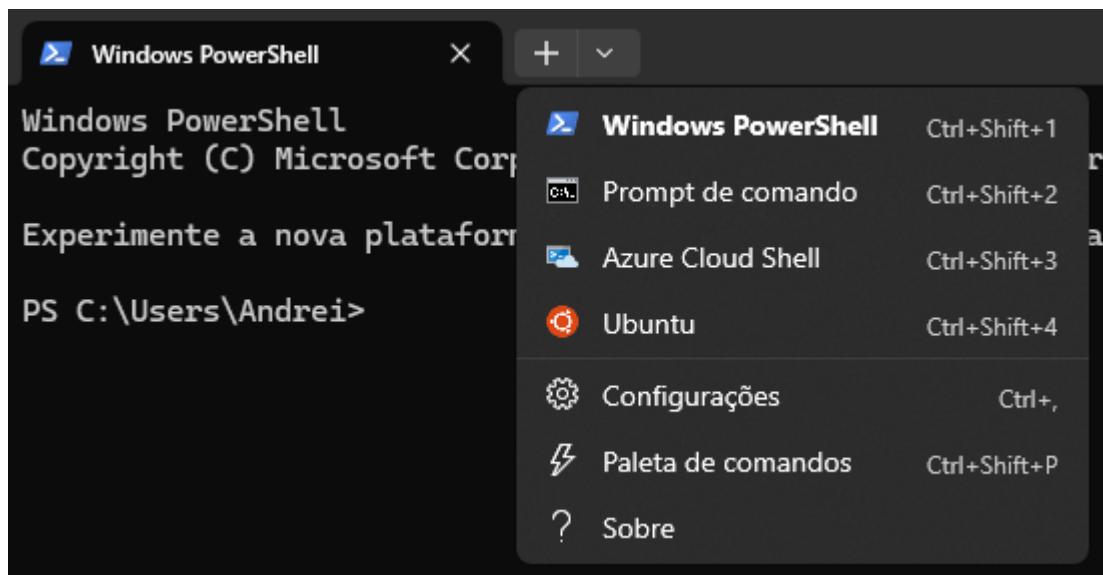
Microsoft Corporation

4.5 ★ | 2.09K classificações

Ferramentas de desenvolvedor

Baixar

Ao abrir o Windows Terminal, o PowerShell será iniciado por padrão, mas você pode selecionar o terminal do Ubuntu clicando na seta ao lado do botão de .



2.5 Instalando o Docker Engine no Ubuntu

Com a nossa distribuição Linux devidamente instalada, a próxima etapa é colocar o Docker para rodar dentro dela. Para isso, basta seguir os passos da [documentação oficial](#) do Docker para o Ubuntu, que estão reproduzidos abaixo.

Passo 1: Adicionar a chave GPG oficial do Docker

Estes comandos, em conjunto, configuram o sistema para adicionar um novo repositório de pacotes (neste caso, o Docker), e garantir que o sistema possa validar a autenticidade dos pacotes baixados desse repositório.



```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
andrei@DESKTOP-05QOAJK:~$ sudo apt-get update
andrei@DESKTOP-05QOAJK:~$ sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyringsom/linux/ubuntu/gpg -o /etc/apt/key
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/key
rings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Passo 2: Adicionar o repositório do Docker à lista de fontes de pacotes do APT

Este comando adiciona o repositório do Docker à lista de fontes de pacotes do APT (o gerenciador de pacotes do Ubuntu), configurando-o para a arquitetura do sistema e verificando os pacotes com a chave GPG armazenada.



```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

```
andrei@DESKTOP-05QOAJK:~$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker
.asc] https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Passo 3: Instalar os pacotes do Docker

Este comando instala o Docker Community Edition, a CLI do Docker, o runtime containerd, além dos plugins Docker Buildx e Docker Compose, equipando o sistema com todas as ferramentas essenciais para criar, gerenciar e executar os contêineres.

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin doc
```



```
andrei@DESKTOP-05QOAJK:~$ sudo apt-get install docker-ce docker-ce-cli conta
inerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dbus-user-session docker-ce-rootless-extras libltdl7 libslirp0 pigz
    slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io dbus-user-session docker-buildx-plugin docker-ce
  docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7
    libslirp0 pigz slirp4netns
0 upgraded, 11 newly installed, 0 to remove and 129 not upgraded.
Need to get 122 MB of archives.
After this operation, 441 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd
  .io amd64 1.7.22-1 [29.5 MB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1
  [63.6 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 dbus-user-se
```

Passo 4: Dar permissão ao usuário atual para rodar o Docker

Este comando adiciona o usuário atual ao grupo docker, permitindo que ele execute comandos Docker sem precisar de privilégios de superusuário (sudo).

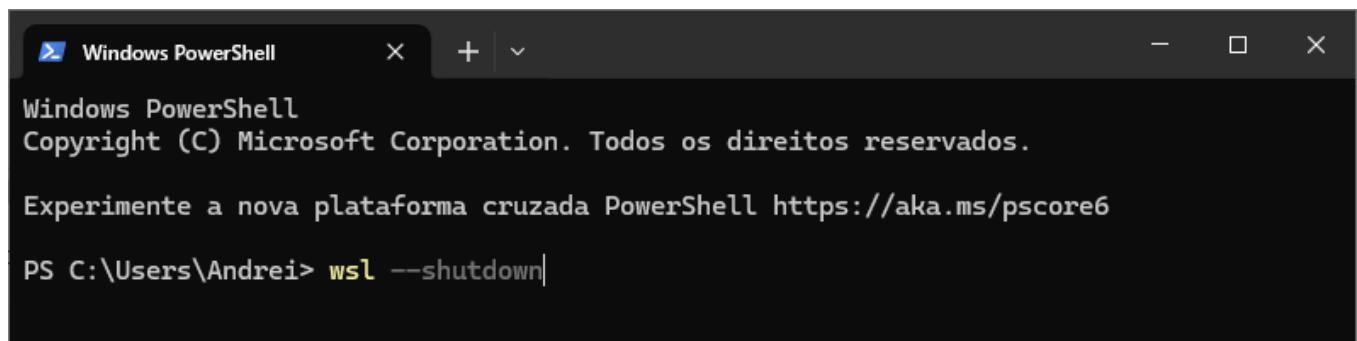
```
sudo usermod -aG docker $USER
```



```
andrei@DESKTOP-05QOAJK:~$ sudo usermod -aG docker $USER
```

Passo 5: Reiniciar o WSL

Fechar o terminal do Ubuntu, e de volta ao PowerShell, executar o comando `wsl --shutdown` para que a mudança tenha efeito.



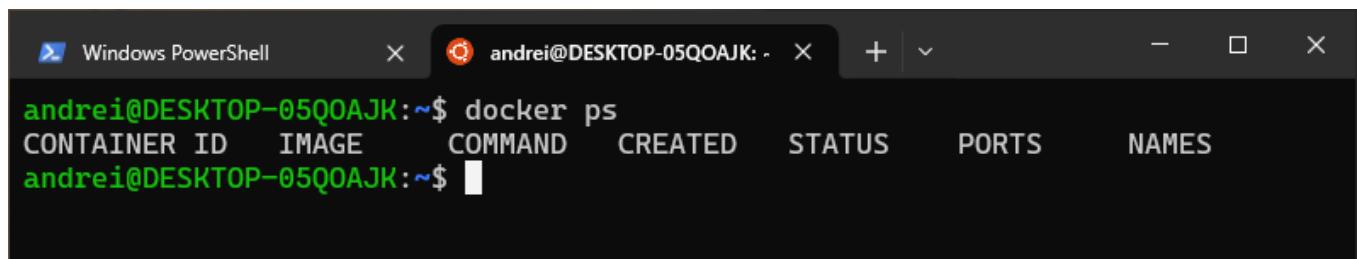
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\Andrei> wsl --shutdown|
```

Passo 6: Verificar Instalação do Docker

Por fim, abrir novamente o terminal do Ubuntu, e executar o comando `docker ps` para verificar que o Docker está rodando com sucesso. O retorno deve ser semelhante ao da imagem abaixo.



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES

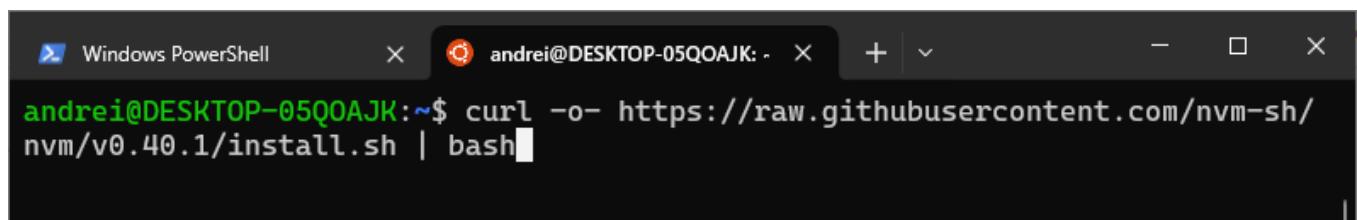
3. Preparando o caminho para o Node.js com o NVM

Com o Docker pronto para uso, vamos partir agora para a instalação do Node Version Manager (NVM), que é a ferramenta que nos ajudará a instalar e gerenciar diferentes versões do Node.js. E isso nós faremos também dentro do WSL, que, daqui pra frente será a nossa "casa" para o desenvolvimento do projeto. O processo é simples e consiste basicamente na execução de um único comando.

Passo 1: Baixar e executar o script oficial

Executar o comando abaixo, disponível no [repositório oficial do NVM](#), para fazer o download e execução do script de instalação.

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
```

```
andrei@DESKTOP-05QOAJK:~$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash|
```

Passo 2: Verificar a Instalação

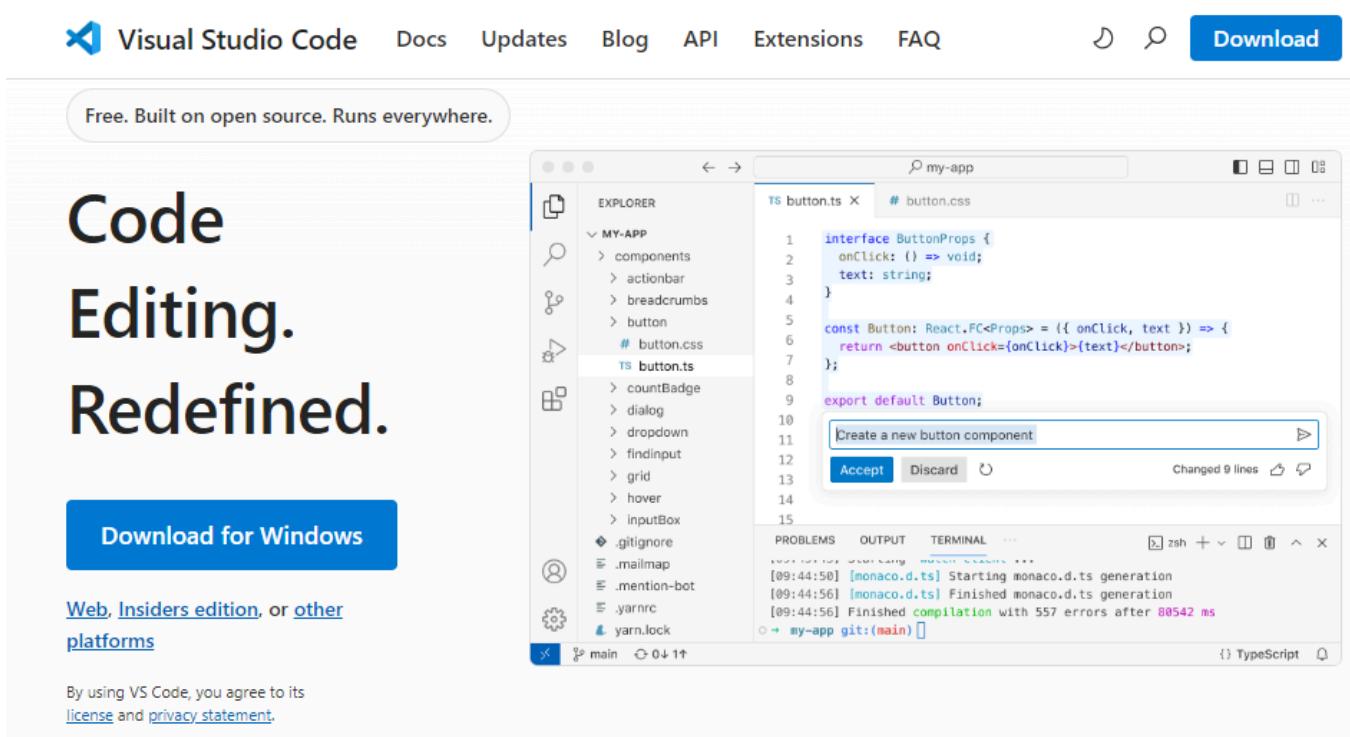
Executar o comando `command -v nvm` a fim de verificar se a instalação foi feita com sucesso. Caso o comando não retorne `nvm`, como na imagem abaixo, fechar e abrir o terminal novamente.

```
Windows PowerShell andrei@DESKTOP-05Q0AJK: ~$ command -v nvm
nvm
andrei@DESKTOP-05Q0AJK: ~$
```

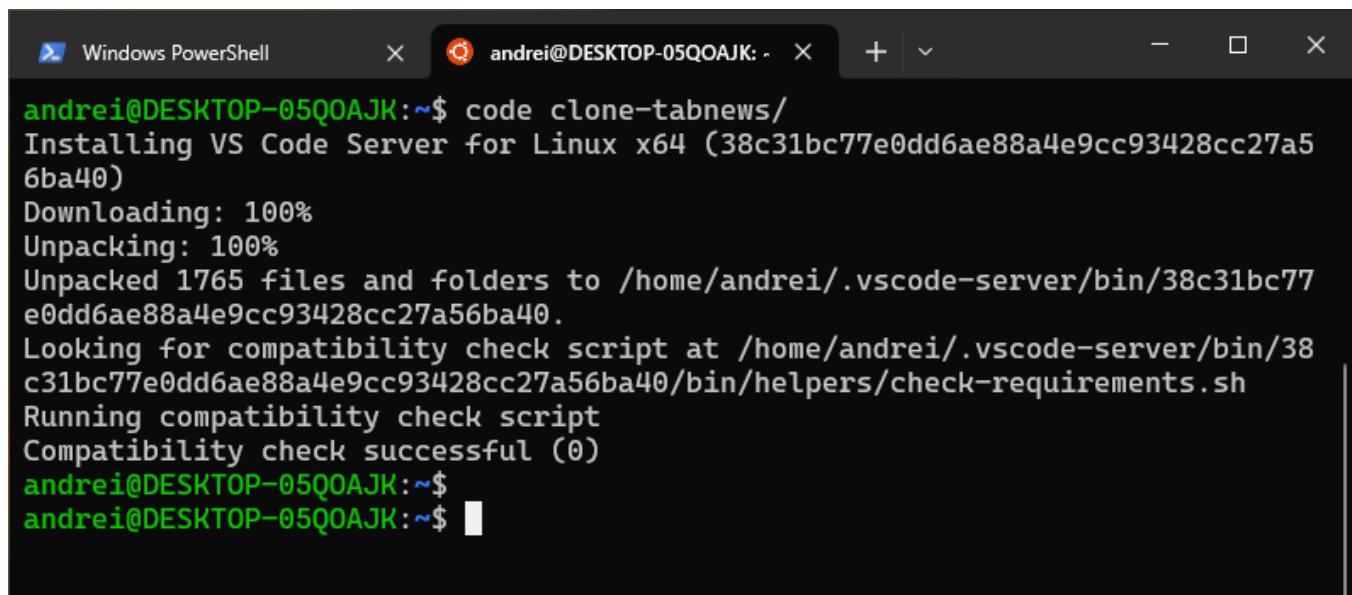
4. Instalando o editor de código

Como editor de código, nós recomendamos a utilização do VSCode. Mas caso você tenha algum outro que seja da sua preferência, sinta-se à vontade para usá-lo.

Para fazer o download do VSCode, basta acessar o [site oficial](#) e clicar no botão `Download for Windows`. Após isso, basta executar o arquivo e seguir os passos do instalador.

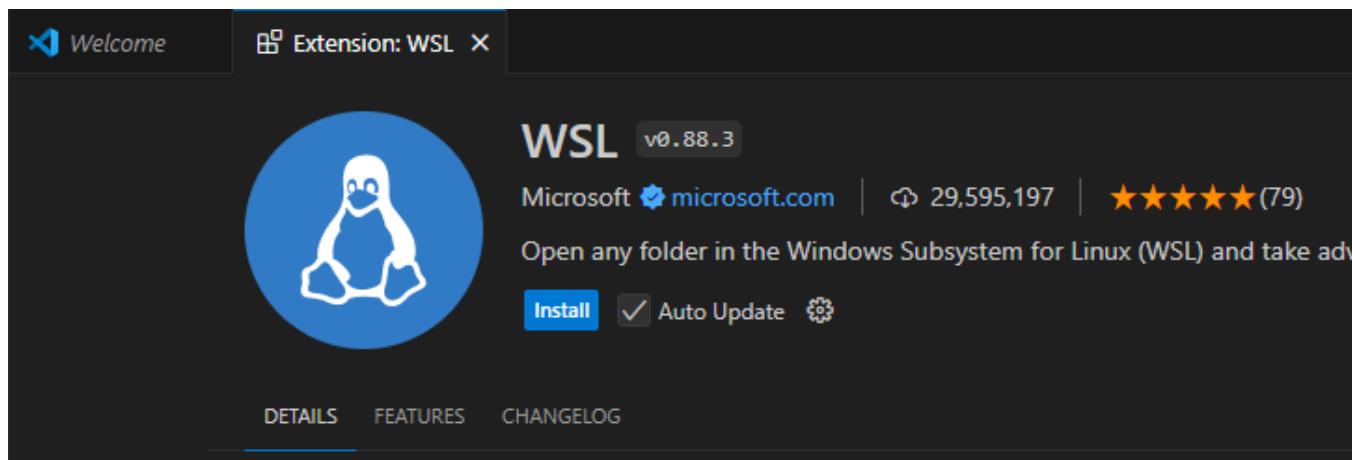


Você pode abrir a pasta do seu projeto no VSCode diretamente do WSL usando o comando `code <nome-da-pasta>` (Obs: vamos ensinar na próxima seção como você pode clonar o repositório do seu projeto no WSL):



```
andrei@DESKTOP-05QOAJK:~$ code clone-tabnews/
Installing VS Code Server for Linux x64 (38c31bc77e0dd6ae88a4e9cc93428cc27a5
6ba40)
Downloading: 100%
Unpacking: 100%
Unpacked 1765 files and folders to /home/andrei/.vscode-server/bin/38c31bc77
e0dd6ae88a4e9cc93428cc27a56ba40.
Looking for compatibility check script at /home/andrei/.vscode-server/bin/38
c31bc77e0dd6ae88a4e9cc93428cc27a56ba40/helpers/check-requirements.sh
Running compatibility check script
Compatibility check successful (0)
andrei@DESKTOP-05QOAJK:~$
andrei@DESKTOP-05QOAJK:~$
```

Isso irá abrir uma janela do VSCode. E para ter uma experiência mais integrada com o WSL, você deve instalar a extensão chamada WSL. Ao abrir o VSCode pela primeira vez, aparecerá um pop up já recomendando a instalação dela. Mas caso não apareça, você pode clicar no ícone de Extensões no menu lateral e buscar por "WSL".



5. Configurando o Git e GitHub

Como o Git já vem instalado por padrão na distribuição Ubuntu, não precisamos nos preocupar em fazer isso. Mas, para que o nosso ambiente fique 100% preparado para realizar os *commits*, é preciso configurar o usuário do Git localmente, e cadastrar as credenciais em nossa conta do GitHub para que possamos enviar as modificações facilmente para o repositório remoto.

5.1 Identificando usuário local do Git

No Git, os commits precisam ser identificados com um nome e um endereço de email. Nós precisamos fazer essa configuração apenas uma vez, e é um processo simples. Basta executar os seguintes comandos:

```
git config --global user.name "Seu Nome"
```



```
git config --global user.email "seu_email@exemplo.com"
```



```
andrei@DESKTOP-05Q0AJK:~$ git config --global user.name "Andrei Alves"
andrei@DESKTOP-05Q0AJK:~$ git config --global user.email "email@example.com"
```

5.2 Configurando o editor padrão do Git

Algumas operações no Git requerem a abertura de um editor de texto. Vamos configurá-lo para que ele use o VSCode como o editor padrão. Para isso, basta executar o seguinte comando:

```
git config --global core.editor "code --wait"
```



5.3 Se conectando com o GitHub com SSH

O protocolo **Secure Shell (SSH)** é uma forma de nos conectarmos a um servidor remoto de modo seguro, usando criptografia. A autenticação da conexão é feita através de um par de chaves (ou senhas), onde uma fica armazenada na máquina do cliente que está buscando a conexão, e a outra no servidor a ser contactado. A vantagem de usar esse sistema é que não precisamos inserir o nosso login e senha sempre que quisermos nos conectar ao GitHub.

Estou assumindo uma instalação zerada do WSL, e portanto que nenhuma chave SSH esteja presente. Caso esse não seja o seu caso, verifique se você já possui uma chave seguindo [essas instruções](#). Se sim, você poderá usá-la ou gerar uma nova seguindo os passos abaixo.

Passo 1: Gerando uma nova chave SSH

Execute o comando abaixo no terminal do Ubuntu, substituindo o email de exemplo pelo que você usa na sua conta do GitHub. Ele irá gerar uma nova chave SSH associada ao email passado.

```
ssh-keygen -t ed25519 -C "seu_email@exemplo.com"
```



Será retornada uma mensagem avisando que a chave está sendo gerada, e uma outra solicitando um caminho e nome para o arquivo da mesma. Basta apertar **Enter** para aceitar a localização padrão.

```
Generating public/private ed25519 key pair.  
Enter a file in which to save the key (/home/usuario/.ssh/id_ed25519):
```



Em seguida, será solicitada uma frase secreta que serve como uma camada de proteção adicional. Você pode ignorar a criação dela apertando **Enter** duas vezes. Se quiser adicionar uma mais tarde, siga **essas instruções**.

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```



Pronto! Par de chaves gerado!

Passo 2: Adicionando a chave ao ssh-agent

O ssh-agent é o programa que irá gerenciar a nossa chave SSH. Com o comando abaixo nós iremos inicializá-lo e deixá-lo pronto para que o terminal possa utilizá-lo para conectar-se ao servidor do GitHub. A mensagem `Agent pid xxxx` será exibida indicando que o programa foi iniciado com sucesso, sendo que o número representado por `xxxxx` é um identificador único do mesmo.

```
eval "$(ssh-agent -s)"
```



Em seguida, execute o comando abaixo para adicionar a chave ao ssh-agent (Obs.: caso a chave tenha sido criada com um nome diferente do padrão, é necessário substituir `id_ed25519` por ele):

```
ssh-add ~/.ssh/id_ed25519
```



Será retornada a mensagem `Identity added: /home/usuario/.ssh/id_ed25519 (seu_email@exemplo.com)`.

Passo 4: Adicionando chave SSH ao GitHub

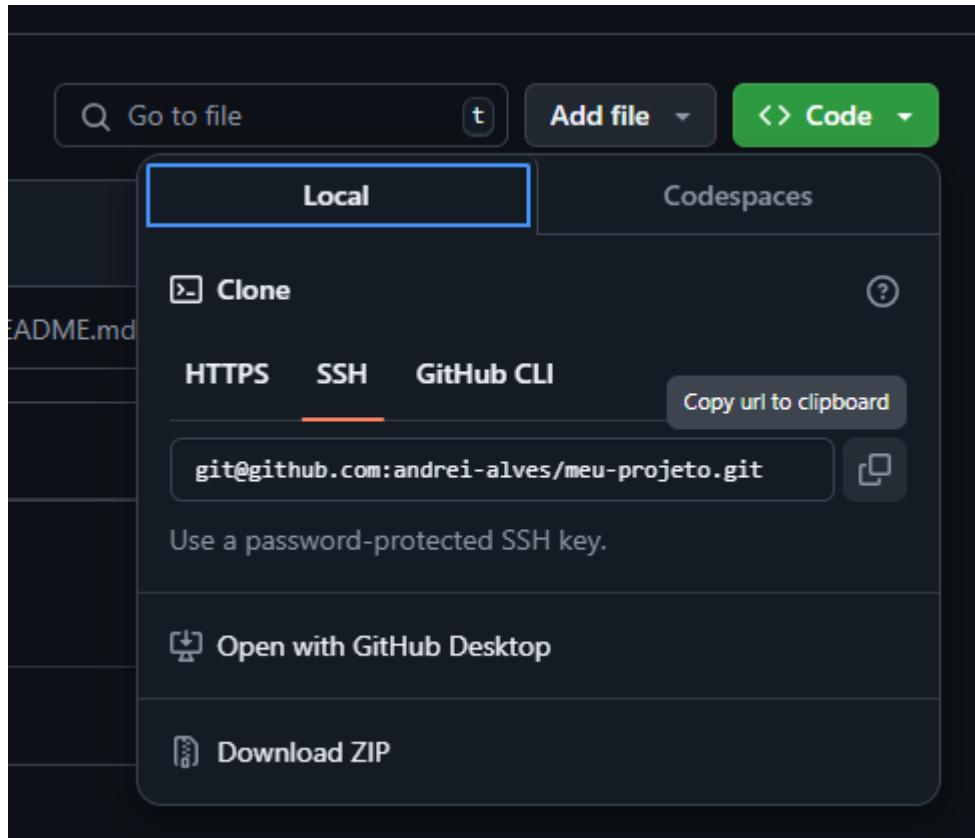
Primeiro, execute o comando `cat ~/.ssh/id_ed25519.pub` a fim de exibir a sua chave no terminal. Depois selecione ela, e copie para a área de transferência.

Em seguida, entre em sua conta do GitHub, e clique na sua foto de perfil no ícone superior direito. No menu aberto clique em `Settings`.

Na seção "Access" da barra lateral esquerda, clique em `SSH and GPG keys`. Em seguida clique no botão `New SSH key` ou `Add SSH key`. Então, no campo "Title" basta adicionar um título para a sua chave (pode ser uma referência ao computador do qual você está acessando), e no campo "Key" colar o conteúdo da chave que foi copiado do terminal. No campo "Key Type"

você pode deixar selecionada a opção "Authentication Key". Por fim, clique em `Add SSH key`, e confirme o seu login, caso seja solicitado.

Com isso, você pode agora clonar o seu projeto para a sua máquina local (no WSL) usando o protocolo SSH, através do comando `git clone`.



Tomando como exemplo o endereço SSH do repositório na imagem acima, o comando a ser executado seria:

```
git clone git@github.com:andrei-alves/meu-projeto.git
```



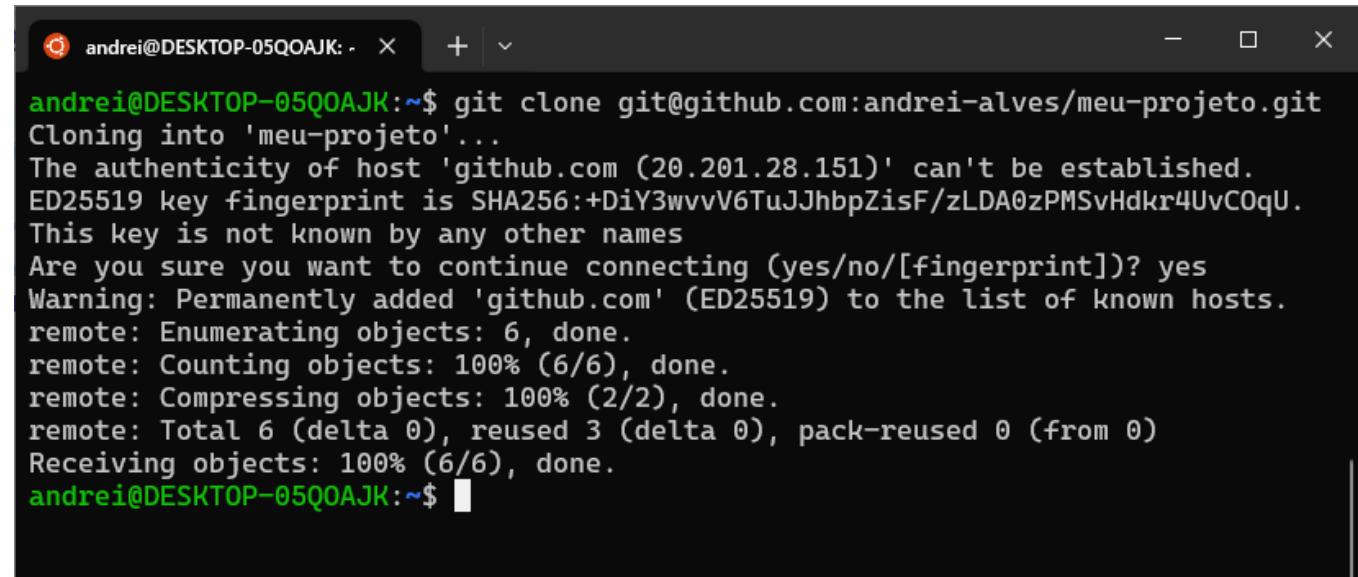
Mas atenção: Na sua primeira conexão, será exibida uma mensagem como a abaixo:

```
The authenticity of host 'github.com (IP ADDRESS)' can't be established.  
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.  
Are you sure you want to continue connecting (yes/no)?
```



Isso acontece porque o GitHub ainda não é um host conhecido pela nossa máquina. E para garantir a autenticidade da chave retornada na mensagem, você pode comparar com a que é disponibilizada na [página oficial](#) (compare com a do tipo Ed25519). Você pode usar [essa ferramenta](#) para comparar as duas chaves, e ao copiar a que é retornada no terminal não inclua o ponto final (por exemplo, na mensagem acima você copiaria SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU).

Ao confirmar a autenticidade, digite `yes` no terminal e aperte `Enter`. E pronto! Agora você pode fazer os seus commits localmente e enviá-los para o seu repositório remoto com facilidade.



```
andrei@DESKTOP-05QOAJK:~$ git clone git@github.com:andrei-alves/meu-projeto.git
Cloning into 'meu-projeto'...
The authenticity of host 'github.com (20.201.28.151)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
andrei@DESKTOP-05QOAJK:~$
```

Para Saber Mais

Documentação Oficial da Microsoft

- [Introdução ao Hyper-V no Windows](#)
- [Documentação do Subsistema Windows para Linux](#)
- [O que é o Subsistema do Windows para Linux?](#)
- [Configurar seu ambiente de desenvolvimento](#)

Documentação Oficial do Docker

- [Instale o Docker Engine no Ubuntu](#) (em inglês)

Footnotes

1. A documentação oficial coloca como requisito apenas **4GB de memória RAM**, o que é relativamente pouco. Todavia, levando em consideração o uso de memória por outros programas, especialmente no processo de desenvolvimento, com editor de código e navegador abertos, essa quantidade é bem limitada. [P](#)
2. Para ser justo, é preciso dizer que, como o **Docker Desktop** possui o **Docker Engine** por debaixo dos panos, ele também permite rodar o Docker via terminal, inclusive no PowerShell. Todavia, usar a linha de comando do Linux para rodar os nossos scripts e outros utilitários garante o máximo de consistência entre os ambientes de desenvolvimento, Integração Contínua, e produção, o que é fundamental na hora de investigar bugs. [P](#)

3. Essa versão mínima requerida para a instalação é de 2020. Caso você possua uma versão ainda inferior é necessário prosseguir com a instalação manual do WSL, que pode ser conferida [nesse tutorial](#). ↗

Concluir aula 20xp

Próxima aula >

Responder

^ Nilo 6 dias atrás

1 Pessoal, o antigravity não tem a extensão WSL da microsoft, isso impede algum procedimento?

Responder

^ Andrei equipe 6 dias atrás

1 Hmm... não saberia dizer com certeza, Nilo . Mas acredito que possa impedir a integração completa.

Responder

^ guilhermevianini 9 dias atrás

1 Galera, estou com problemas com o ubuntu. Eu já tinha instalado uma versão há algum tempo atrás e acredito que as duas agora estão em conflito (consigo pressupor isso porque no terminal aparecem dois ícones do ubuntu).

Não sei exatamente o que fazer porque jurava ter desinstalado o antigo.

Responder

^ Andrei equipe 8 dias atrás

1 Opa, Guilherme! Qual seria o conflito que está acontecendo? Está aparecendo alguma mensagem de erro?

Responder

^ guilhermevianini 6 dias atrás

1 Não há mensagem de erro, simplesmente não inicia a parte do login no terminal.

[Responder](#)[Ver mais 1 resposta](#)

^ [ayrtonmoura](#) 18 dias atrás

2 olá pessoal, eu sou novo e não estou acostumado com isso mas decidi tentar, e já instalei o Ubuntu mas o sudo apt-get update só falha

```
Ign:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Err:1 http://security.ubuntu.com/ubuntu noble-security InRelease
    Connection failed [IP: 91.189.92.22 80]
Err:2 http://archive.ubuntu.com/ubuntu noble InRelease
    Connection failed [IP: 185.125.190.82 80]
Err:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
    Connection failed [IP: 91.189.91.83 80]
Err:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
    Connection failed [IP: 91.189.91.82 80]
Reading package lists... Done
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/noble/InRelease Connection f
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/noble-updates/InRelease Conn
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/noble-backports/InRelease Co
W: Failed to fetch http://security.ubuntu.com/ubuntu/dists/noble-security/InRelease Co
W: Some index files failed to download. They have been ignored, or old ones used instead
```



não sei mais o que fazer, já troquei o dns para o do google 8.8.8.8 já mudei o ip mirror para os servidores brasileiros enfim não sei mais o que fazer

[Responder](#)

^ [ayrtonmoura](#) 18 dias atrás

2 consegui resolver com sudo sed -i 's|http://|https://|g' /etc/apt/sources.list.d/* a rede bloqueava http

[Responder](#)[^ Andrei](#) equipe 18 dias atrás

1 Opa, Ayrton! Parabéns por ter conseguido resolver e obrigado por compartilhar a sua solução. 🤘

[Responder](#)[^ Fox](#) 27 dias atrás

7 WSL + Ubuntu + Docker: um detalhe simples que fez toda a diferença

Pessoal,

Quero compartilhar uma dificuldade que tive ao configurar WSL + Ubuntu + Docker, causada por um detalhe simples, mas importante.

Após instalar o WSL, o Ubuntu e o Docker, eu:

- Abri o VSCode pelo Windows
- Criei a pasta do projeto pelo Windows Explorer

Apesar de tudo parecer correto, comecei a ter problemas (paths, permissões, Docker, etc.).

O ponto que eu não tinha entendido bem é que o **WSL possui seu próprio sistema de arquivos**, separado do Windows. Na prática, ele funciona como se fosse um ambiente Linux “à parte” (não é uma partição de fato, mas o comportamento é parecido).

Quando criamos o projeto no Windows (`c:\...`) e acessamos pelo WSL (`/mnt/c/...`), estamos usando um *filesystem* montado. Isso até funciona, mas **não é o fluxo recomendado**, principalmente quando envolve Docker e ferramentas que assumem comportamento Linux nativo.

O que resolveu

Passei a fazer tudo dentro do WSL (olha a foto de como é o terminal aqui no tutorial):

```
cd ~  
mkdir meu_projeto # Isso cria a pasta do projeto (make dir - crie um diretório)  
cd meu_projeto # Isso vai para a pasta do projeto (change directory)  
code . # Isso aqui abre o VSCode
```



Assim:

- O projeto fica em /home/usuario/...
- O VSCode abre conectado ao WSL
- Paths, permissões e Docker funcionam corretamente

Fica a dica

Para quem usa WSL com Docker:

- Crie o projeto no filesystem do WSL
- Abra o VSCode a partir do WSL

Depois que ajustei isso, os problemas desapareceram.

Espero que ajude alguém que esteja passando pela mesma situação.

[Responder](#)

^ Andrei equipe 26 dias atrás

1 Excelente dica, Fox ! Eu meio que deixo isso implícito no tutorial, mas acredito que preciso melhorar o texto para deixar essa questão mais clara, porque outros alunos também cometem esse mesmo erro inicial que você teve. Muito obrigado por compartilhar aqui! 

[Responder](#)

^ dudu2701 29 dias atrás

1 Consegi Seguir o passo a passo com sucesso, mas estou com dificuldades para conseguir rodar o npm run dev. Consegi achar a pasta do meu projeto dentro do ubuntu, mas quando tento rodar aplicação ocorre esse erro:

```
PS Microsoft.PowerShell.Core\FileSystem::\wsl.localhost\Ubuntu\home\dudu\clone-tabnews> npm run dev
npm : O arquivo C:\Program Files\nodejs\npm.ps1 não pode ser carregado porque a execução de scripts foi desabilitada neste sistema. Para obter mais informações, consulte about_Execution_Policies em https://go.microsoft.com/fwlink/?LinkId=135170.
```

No linha:1 caractere:1

- npm run dev
- + CategoryInfo : Erro de Segurança: (:) [], PSSecurityException



```
+ FullyQualifiedErrorId : UnauthorizedAccess
```

PS Microsoft.PowerShell.Core\FileSystem::\wsl.localhost\Ubuntu\home\dudu\clone-tabnews>

Alguém consegue me auxiliar?

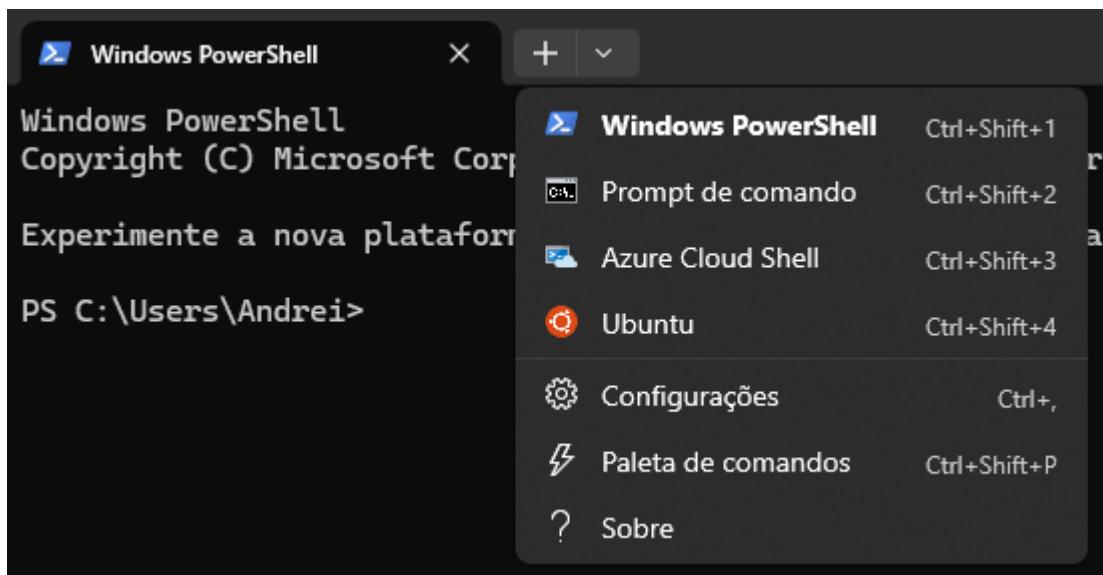
Responder

Andrei equipe 29 dias atrás

1 Opa, dudu2701 ! Que bom que você conseguiu seguir o passo a passo! 🎉

Vejo que você está utilizando o PowerShell, mas o ideal é que você utilize o terminal do Ubuntu do seu WSL.

Ao abrir o Windows Terminal (que eu recomendo instalar no tutorial), o PowerShell será iniciado por padrão, mas você pode selecionar o terminal do Ubuntu clicando na seta  ao lado do botão de :



Responder

dudu2701 29 dias atrás

1 A certo obrigado, eu estava tentando usar o terminal dentro do próprio vscode, mas ainda estou tendo dificuldades para conseguir rodar o meu código mesmo rodando no ubuntu

Não consegui encaminhar uma imagem, mas vou mandar o trecho que deu erro ao rodar o `npm run dev`:

```
dudu@LAPTOP-QHQRKI47:~/clone-tabnews$ npm run dev
```

```
> clone-tabnews@1.0.0 dev  
> next dev  
  
sh: 1: next: not found  
dudu@LAPTOP-QHQRKI47:~/clone-tabnews$
```

Eu fiz algumas alterações, mas excluir a pasta do clone-tabnews e fiz o clone novamente, mas ainda não estou conseguindo executar o projeto

[Responder](#)[Ver mais 3 respostas](#)

^ BrB 1 mês atrás

2 O mais maravilhoso é que nao vou perder essas instruções no futuro pois o curso é vitalício, então nao preciso me preocupar:)

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Sim, BrB ! Você pode fazer as aulas sem pressa! 🙌

[Responder](#)

^ SandovalBezerra 1 mês atrás

1 ja fiz a instalação do WSL via chocolatey em uma de minhas maquina, e o wsl tem versões, qual e a que devemos instalar o wsl2? outra posso usar o chocolatey ou é imprescindível para o curso seguir como esta a instalação aqui??

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Seria o WSL2, Sandoval. 🤝

outra posso usar o chocolatey ou é imprescindível para o curso seguir como esta a instalação aqui??

Você pretende usar o chocolatey apenas para instalar o WSL? Embora eu não tenha testado, acredito que você pode usar ele também, já que você fez uma vez e deu certo.

[Responder](#)

^ viturino 1 mês atrás

1 O meu Windows 11 Pro não quis acordo com a instalação do Ubuntu kkkk tenho que desativar umas coisas de segurança na BIOS se não me engano, vou seguir com o Codespaces mesmo, e eu vou endoidar é kkkkk

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Sem problemas, viturino ! Nesse caso, eu recomendo que você continue com o Codespace mesmo. 🤗

[Responder](#)

^ Kikows 1 mês atrás

1 Se ajudar, no Windows Search Bar, pesquise por: Windows features
Aí, seleciona as seguintes features e dê restart ao PC.

- Hyper-V
- Virtual Machine Platform
- Windows Hypervisor Platform
- Windows Subsystem for Linux

Se mesmo assim o WSL não abrir, desinstale e volte a instalar o WSL.

[Responder](#)

[↳ Ver mais 1 resposta](#)

^ Ignacio02 1 mês atrás

1 Uma dúvida de iniciante após terminar todos os passos:
Agora, sempre que eu criar um projeto pelo VSCode, ele vai subir automaticamente para o Github? Como isso vai funcionar na prática?

[Responder](#)

^ Felpx 1 mês atrás

3 ele não sobe automaticamente, ignacia. Voce tem que fazer o que chamamos de commit, que seria a grosso modo enviar o seu projeto para o github.

[Responder](#)

^ Andrei equipe 1 mês atrás

2 Opa, Ignacio02 ! Como o Felpx , esse upload não vai ser automático. O que nós fizemos aqui nesta aula foi apenas criar uma cópia do nosso repositório na nossa máquina local e estabelecer uma conexão segura com o GitHub.

Nas próximas aulas é que o Filipe vai mostrar como a gente pode enviar as alterações que nós fazemos no nosso repositório local para o GitHub. 

[Responder](#)

[↑ Ver mais 1 resposta](#)

^ Braerc 1 mês atrás

1 Olá, tudo bem? fiz todo o passo a passo mas não tem nenhuma pasta do repositório no meu pc condizente com o projeto que estou desenvolvendo, isso pode gerar algum problema na hora de codar? ou é indiferente? basta estar alinhado meu ubuntu com o git?

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Opa, Braerc ! Não sei se entendi muito bem a sua dúvida. Você procurou a pasta do seu projeto no sistema de arquivos do Windows e não achou? Foi isso?

[Responder](#)

^ Braerc 1 mês atrás

1 Olá, bom dia, não, eu fiz todo passo a passo da conexão do windows com ubuntu + github so que me bateu uma curiosidade como que eu linkei meu repositório ao ubuntu+windows sem algo local ex: uma pasta, pois na minha cabeça a conexão seria, pasta do repositório no meu pc -> conexão no ubuntu -> conexão com o github e ai sim tudo estaria integrado, não consegui compreender a conexão sem um ambiente local, caso não tenha ficado claro pode pedir para eu reescrever sem problemas e obrigado pela atenção!!!

[Responder](#)

[↑ Ver mais 1 resposta](#)

^ SauloFelipe 1 mês atrás

1

Fiz a configuração mas vou explorar o ambiente em nuvem do github, o que é melhor para iniciantes como eu ?

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Saulo, se você conseguiu configurar o seu ambiente local com sucesso, então você pode continuar com ele. 🤝

[Responder](#)

^ SoftMissT 1 mês atrás

1 Pera, eu tenho que fazer isso mesmo usando um vscode, tipo o antigravity?

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Opa, SoftMissT ! Acredito que o Antigravity seja apenas um editor de código integrado com a IA, correto? Você ainda precisaria do Docker. 🤝

[Responder](#)

^ BiancaFarias 1 mês atrás

1 Não sei se vou ter coragem de fazer outra rota por hora 😅

[Responder](#)

^ Andrei equipe 1 mês atrás

2 Sem problemas, Bianca. Você pode seguir pelo Codespaces. 🤝

[Responder](#)

^ billdev 1 mês atrás

1 Olá, o NVM pode ser instalado e configurado na criação de uma image ou nesse caso é importante estar na raiz do WSL?

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Não entendi a sua dúvida, billdev . 😕

[Responder](#)

^ billdev 1 mês atrás

1 Quando criamos uma imagem no Docker podemos aplicar as configurações padrão que vai ser executada no container, uma dessas configurações pode ser o NVM. A pergunta, o recomendado do NVM é estar no WSL ou pode ser configurado na Imagem que vai montar o container?

[Responder](#)

[↑ Ver mais 1 resposta](#)

^ DenisAguiar06 1 mês atrás

2 Deu tudo certinho, achei legal a ideia de mesmo sendo iniciante me desafiar a novos desafios.

[Responder](#)

^ Andrei equipe 1 mês atrás

◀ Boooa, Denis!!! Fico feliz que tenha conseguido!

[Responder](#)

^ MaiconDante 1 mês atrás

2 Conseguí, porém tive que entrar na BIOS e ativar o SVM que estava desabilitado sobre virtualização de máquinas e após reiniciar e fazer procedimentos conforme tutorial acima, consegui e tudo funcionando normalmente.

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Maravilha, Maicon! 🎉

[Responder](#)

^ ThiagoFranchin 1 mês atrás

2 Tudo certo, deu bom, só fiz o passo do @Joaolucs08 aqui dos comentários a mais. Valeu a todos!!!

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Show de bola, Thiago!

[Responder](#)

^ theviniciusgomes 2 meses atrás

2 Deu tudo certo, devido ao tempo limitado acabei levando três dias, mas conclui. Apesar de eu ter decidido utilizar o codespaces queria fazer essa configuração... Gosto de mexer em linhas de comando e fazer essas configurações. Agora vou me meter com o Arch Linux. hhehehe

[Responder](#)

^ Andrei equipe 1 mês atrás

2 Boa, Vinicius! Manda bala! 🤘

[Responder](#)

^ Joaolucs08 2 meses atrás

6 Fala pessoal!

Queria compartilhar que deu tudo certo comigo, mas gostaria de fazer uma observação importante para quem está começando.

Cenário: Você executou o wsl --install mas o Ubuntu ainda não apareceu no menu iniciar do Windows.

Provável causa: O WSL foi instalado, mas a distribuição Ubuntu não foi.

Como verificar:

Abra o PowerShell como administrador e execute:

```
wsl --list --verbose
```

Se estiver instalado corretamente, você verá a lista de distribuições.

Se não aparecer nada (como foi meu caso), basta instalar o Ubuntu especificamente com o comando:

```
wsl --install -d Ubuntu
```

Depois disso, você já deve conseguir encontrar o Ubuntu no menu iniciar e também no botão "+" dentro do Terminal do Windows!

Vlw pessoal ! 🌟

[Responder](#)

^ ZioNN 1 mês atrás

1 Rapaz, estava aqui quebrando a cabeça porque não encontrava o Ubuntu. Atualizem essa questão no curso, pessoal. Valeu!

[Responder](#)

^ Andrei equipe 1 mês atrás

1 Maravilha, João! Muito obrigado pela contribuição! ❤️

[Responder](#)

[↑ Ver mais 5 respostas](#)

^ coutinho 2 meses atrás

1 Estou com um problema ao tentar usar o comando `code <nome_do_arquivo>` porem ele me da este aviso

Command 'code' not found, but can be installed with:
sudo snap install code



e ao dar o comando ele aparece outra mensagem:

error: This revision of snap "code" was published using classic confinement and the arbitrary system changes outside of the security sandbox that snaps are usually which may put your system at risk.

If you understand and want to proceed repeat the `command` including `--classic`.



e se eu instalar usando o `--classic` ele abre um vs code dentro do wsl ao inves do que eu instalei no windows.

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Opa, coutinho ! Dá uma conferida nessa thread de comentários [aqui](#), onde o aluno estava tendo o mesmo problema. Especialmente nessa minha resposta [aqui](#), que foi a

definitiva para resolver. 🤪

[Responder](#)

^ CarlosFast 2 meses atrás

3 Atenção galera que usa placas da Gigabyte!

Após instalar o WSL e tentar iniciar a instalação do Ubuntu, pode acontecer de vocês esbarrarem neste erro (ou similar):

Installing, this may take a few minutes... WslRegisterDistribution failed with error: 0x80370102 Please enable the Virtual Machine Platform Windows feature and ensure virtualization is enabled in the BIOS. For information please visit <https://aka.ms/enablevirtualization> Press any key to continue...

Nestes casos, é necessário ativar a Virtualização da placa mãe através da BIOS - nos casos das placas modernas da Gigabyte, conhecido como SVM Mode.

Como os modelos variam bastante, e portanto os menus das BIOS, recomendo buscar como ativar especificamente para o modelo de suas placas mães.

Pra quem não tem proximidade com essas coisas, recomendo não se aventurar demais lá dentro dos menus de BIOS e focar exclusivamente na ativação da SVM Mode.

[Responder](#)

^ Filho 16 dias atrás

1 Eu vou de Codespaces por enquanto.

[Responder](#)

^ Andrei equipe 2 meses atrás

2 Perfeito, Carlos! Muito obrigado pela dica! 🤪

[Responder](#)

^ biaginibe 2 meses atrás

2 Ótima explicação!

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Fico feliz que tenha gostado, biaginibe ! 😊

[Responder](#)

^ ViniYatsuda 2 meses atrás

1 Please enable the Virtual Machine Platform Windows feature and ensure virtualization is enabled in the BIOS.

Estou com esse erro quando instalo o WSL, não consegui entender como arrumar isso :(

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Vini, eu encontrei esses passos abaixo num [fórum da Microsoft](#). Experimente executá-los para tentar resolver o problema.

Geralmente, o problema ocorre por corrupção no Hyper-V. Reinstalar o recurso costuma resolver.

Passo 1 — Desinstalar

1. Abra o menu Iniciar e pesquise por “Ativar ou desativar recursos do Windows”.
2. Desmarque “Hyper-V” e “Plataforma de Máquina Virtual”.
3. Clique em OK e reinicie.

Passo 2 — Reinstalar

1. Abra novamente “Ativar ou desativar recursos do Windows”.
2. Marque “Hyper-V” e “Plataforma de Máquina Virtual”.
3. Clique em OK e reinicie.

Passo 3 — Ativar o hypervisor na inicialização

1. Abra o menu Iniciar e procure por Prompt de Comando.
2. Clique com o botão direito e selecione Executar como administrador.
3. Execute:

```
bcdedit /set hypervisorlaunchtype Auto
```



4. Reinicie o computador e tente instalar o Ubuntu novamente.

[Responder](#)

^ ELIANDRO 2 meses atrás

1 se eu formatar o note e colocar linux seria um caminho ideal digamos assim?

[Responder](#)

^ Andrei **equipe** 2 meses atrás

1 Seria o ideal apenas se você quer ficar exclusivamente com Linux, Eliandro.

A vantagem de usar o WSL é que nós podemos trabalhar com os dois sistemas simultaneamente. 🤝

[Responder](#)

^ DFIKEDA 2 meses atrás

2 Andrei, ao tentar instalar em meu micro, e eu já tenho o WSL instalado ele aparece a lista de opções da lib. Isso faltou no passo a passo, pois apesar de aparecer a "ajuda" não diz que o WSL já está instalado kkkkkkkkk Fui encontrar na documentação oficial e acho que é interessante ajudar quem se encontrar com o mesmo "esquecimento" que eu tenho rsrs.
<https://learn.microsoft.com/pt-br/windows/wsl/install>

[Responder](#)

^ Andrei **equipe** 2 meses atrás

2 Ah, perfeito, DFIKEDA ! Obrigado pela dica! 🤝

[Responder](#)

^ DFIKEDA 2 meses atrás

1 Tive um problema, no passo 2.6:

docker ps

failed to connect to the docker API at unix:///var/run/docker.sock; check if the path is correct and if the daemon is running: dial unix /var/run/docker.sock: connect: no such file or directory

Apesar de eu ter procurado... não encontrei uma solução. Depois eu volto aqui e tento reinstalar a Distro pra ver se corrige.

[Responder](#)

^ sylar 1 mês atrás

1 Infelizmente estou enfrentando o mesmo problema, mas gostaria de resolvê-lo se possível, visto que pretendo utilizar o docker em outros projetos e rodar de forma local...

Edit 1:

Bom, depois de quebrar muito a cabeça, e muito provavelmente por incompatibilidade com versão mais antiga do Windows (tive vários problemas pra instalar o wsl2 com ubuntu por causa disso também), o docker desktop resolveu pra mim... não era a opção que eu queria mas, melhor que ficar travado aqui por mais tempo

Edit 2:

No fim das contas, consegui instalar o Docker Engine pelo bash (precisa desinstalar o Docker Desktop ou pelo menos não abri-lo enquanto roda o bash do linux), fazendo todas as instalações de updates do windows que estavam pendentes a alguns anos kkk agora me pergunto como será para as pobres almas do w10 que perderam o suporte e essas atualizações.

Mesmo com w10, de alguma forma ainda consegui, então há esperança.

[Responder](#)

^ Andrei equipe 2 meses atrás

1 DFIKEDA , por acaso você já tinha o Docker Desktop instalado na sua máquina?

[Responder](#)

[↳ Ver mais 2 respostas](#)

^ Richardves 2 meses atrás

2 Todo certo até aqui haha

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Show de bola! 😱

[Responder](#)

^ luizfbp 2 meses atrás

1 Vou manter o Docker Desktop no Windows já que ele é integrado com WSL, espero não ter dor de cabeça kkkkkk

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Opa, Luiz! Se você já tem o Docker Desktop configurado, pode seguir com ele mesmo sem problemas. 🤗

[Responder](#)

^ luizfbp 2 meses atrás

1 Sim, sim. Mas não duvido encontrar algum problema por tá usando assim, normal de acontecer kkkkkkk. Obrigado por responder.

[Responder](#)

[↑ Ver mais 1 resposta](#)

^ matheusqueirozds 2 meses atrás

1 Muito bom. Deu tudo certo, por aqui.

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Maravilha, Matheus! Fico feliz que tenha conseguido! 🎉

[Responder](#)

^ RenanGabriel47 2 meses atrás

1 de primeira não sei se tinha dado certo pois não apareceu a mensagem

The authenticity of host 'github.com (IP ADDRESS)' can't be established.

ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.

Are you sure you want to continue connecting (yes/no)?

mas olhando a comunidade pessoas que tiveram o mesmo problema consegui resolver, bem facil

já estava conectado!

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Show de bola, Renan! Que bom que conseguiu resolver! 🎉

[Responder](#)

Wilpersan 2 meses atrás

1 Estou com dificuldades de seguir tudo sem nenhum problema, é minha total primeira vez com isso optei pelo Linux por ser um objetivo antigo, mas confesso que estou com uma curva de aprendizado muito grande. Seguindo os passos, fiz pelo Linux e cheguei na configuração do Git, na seção 5 onde se lê "Mas atenção: Na sua primeira conexão, será exibida uma mensagem como a abaixo:", ao invés da msg informada eu recebi "atal: could not create work tree dir 'wilpersan_tabnews': Permissão negada".

Conseguem me dar um help?

Grato

[Responder](#)

Andrei equipe 2 meses atrás

1 Opa, Wilpersan ! Tudo joia? Esse é um erro de permissão, e provavelmente você está tentando clonar a pasta do seu repositório em um diretório que você não tem permissão, como o diretório root, por exemplo. Antes de executar o comando `git clone`, experimente executar `cd ~` para ir para o diretório do seu usuário. Depois, tente seguir o passo a passo novamente. 🍍

[Responder](#)

arthurmartinnss 2 meses atrás

2 É obrigatório a utilização da SSH key no github? acabei não vinculando ainda a minha key gerada (deixei guardada), pois estou usando o github para o meu estágio e fiquei com receio de gerar conflito caso eu tente dar commit etc, estando em outra máquina... no caso eu irei precisar criar uma SSH key no PC da empresa e vincular ela também, correto?

[Responder](#)

Andrei equipe 2 meses atrás

1 É obrigatório a utilização da SSH key no github?

Opa, Arthur! Não é obrigatório, mas facilita bastante o processo de mandar commits para o GitHub.

estou usando o github para o meu estágio e fiquei com receio de gerar conflito caso eu tente dar commit etc, estando em outra máquina... no caso eu irei precisar

| criar uma SSH key no PC da empresa e vincular ela também, correto?

Como você está se conectando ao GitHub no PC da empresa? Você pode configurar diferentes chaves para a sua conta em diferentes PCs. 🤔

[Responder](#)

^ LucaOuriques 2 meses atrás

2 Difícilmente costumo conseguir fazer tudo funcionar de primeira apenas lendo a documentação...

Mas essa documentação está muito bem escrita, consegui configurar tudo de primeira sem problemas nenhum!

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Maravilhaaa, Luca!!! Fico feliz com isso!!! 🎉

[Responder](#)

^ FelipeMoreira 2 meses atrás

1 Boa documentação, deu tudo certo aqui, porém no final quando clonei o repositório não apareceu essa mensagem:

"Mas atenção: Na sua primeira conexão, será exibida uma mensagem como a abaixo:

The authenticity of host 'github.com (IP ADDRESS)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
Are you sure you want to continue connecting (yes/no)?"

Apenas clonou o repositório direto, isso pode gerar algum problema no futuro?

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Opa, Felipe! Fico feliz que tenha gostado! ❤️

| Apenas clonou o repositório direto, isso pode gerar algum problema no futuro?

Provavelmente não. Você já tinha se conectado ao GitHub alguma vez na sua máquina via SSH?

Vamos tentar o seguinte: para verificar se o GitHub está registrado como um host conhecido, execute o comando `ssh-keygen -F github.com`. Se não estiver, o retorno deve ser semelhante a esse:

```
Host github.com not found in ~/.ssh/known_hosts
```

[Responder](#)

^ FelipeMoreira 2 meses atrás

2 Você estava certo, rodei o comando `ssh -T git@github.com` para verificar a conexão e apareceu.

[Responder](#)[Ver mais 1 resposta](#)

^ RenanVicente 2 meses atrás

1 Não vou utilizar, mas esta muito bem documentado!

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Show! Que bom que você gostou, Renan! 🤪

[Responder](#)

^ vitorcarvalho 2 meses atrás

2 Bem tranquilo de seguir o passo a passo

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Fico feliz que tenha achado fácil, Vitor! 🎉

[Responder](#)

^ MaxwellRyan 2 meses atrás

2

Pensei que teria muitos problemas a fazer, mas os passos a seguir não são complexos, bem simples e fáceis. 😊

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Opa! Fico feliz que você tenha achado fácil, MaxwellRyan ! Valeu pelo feedback! 🤍

[Responder](#)

^ MarlonGiovany 2 meses atrás

2 Por enquanto ficare no Codespaces, mas logo mais estarei de volta aqui nessa aba

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Sem problemas, Marlon! Manda bala! 💪

[Responder](#)

^ JhonnyAlves 2 meses atrás

2 Resultado com sucesso! vamo que vamo :)

[Responder](#)

^ Andrei equipe 2 meses atrás

1 Showw de bola, Jhonny!!! Fico feliz que tenha conseguido! 😃 🎉 🎉

[Responder](#)

^ MardoqueuLS 3 meses atrás

2 Parabens Andrei, excelente tudo. Meu ambiente é linux mas consegui finalizar a configuração do Git seguindo aqui tb. Como eu inicio o projeto no VS Code?

UPDATE: CONSEGUI KKKK. Fica a pasta do projeto na raiz, acesse ela pelo terminal, dá um code . e o projeto inicia dentro da pasta abrindo o VsCode. Fazia tempo que eu não fazia isso, tinha esquecido.

[Responder](#)

^ Andrei equipe 3 meses atrás

1 Maravilhaa, Mardoqueu!!! Fico feliz que tenha gostado! 😊

[Responder](#)

[Ver mais 46 respostas \(422 ocultas\)](#)

© 2026 curso.dev

[Contato](#) [Termos de Uso](#) [TabNews](#)