

# Terminal no Windows

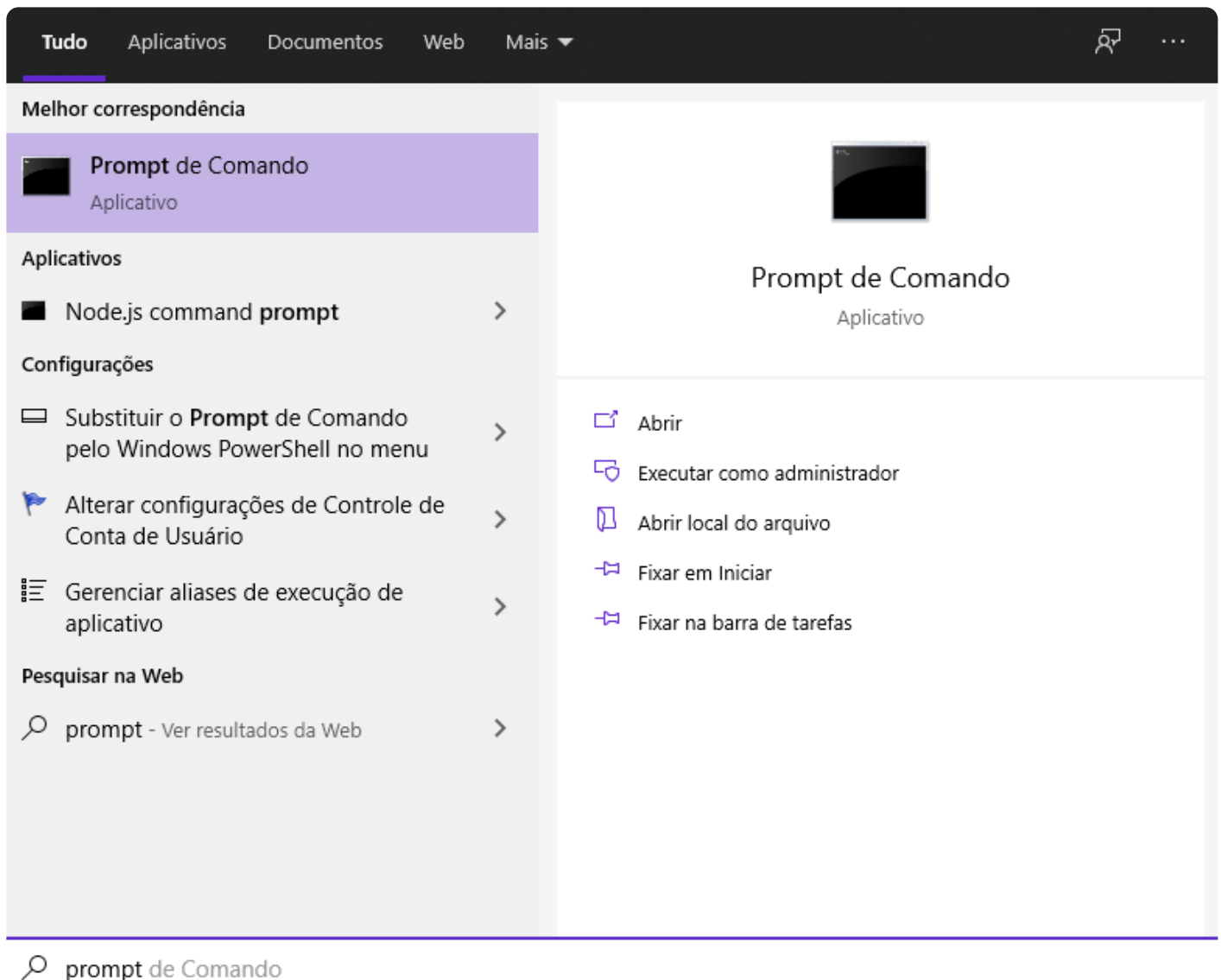
O Terminal no Windows é uma ferramenta imprescindível na vida dos profissionais da TI, do programador ao profissional de redes, pois é uma ferramenta capaz de fazer instalações, configurações, checar dados e navegar pelo sistema. O terminal não é uma novidade para ninguém no mundo da programação, ele sempre esteve ali e sempre estará, no entanto, não são todos que gostam ou sabem usá-lo. Caso esse seja seu caso, vamos te ajudar nesse artigo.

## CMD vs PowerShell

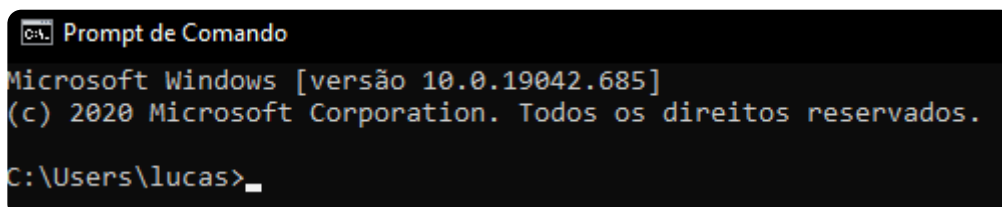
Atualmente, encontramos no **Windows** dois terminais diferentes. O **CMD** (ou Prompt de Comando) é o mais clássico, com fundo preto. Já o **Windows PowerShell** é mais recente e tem o fundo azul. Em resumo, o PowerShell é uma evolução do CMD, fazendo tudo o que o CMD faz e um pouco mais. Como por exemplo, usar alguns comandos próprios de sistemas baseados em Unix, ou seja, Linux e MacOS.

## Conhecendo o CMD

Para abrir o CMD digite “CMD” ou “Prompt” no menu Iniciar:



Quando aberto dessa forma ele apontará diretamente para a pasta do seu usuário. Em outras palavras, no início o CMD vai apontar para o seguinte caminho: **C:**, que é o HD principal da sua máquina, onde o Windows está instalado; e **Users**, que é a pasta de usuários e seu nome de usuário.



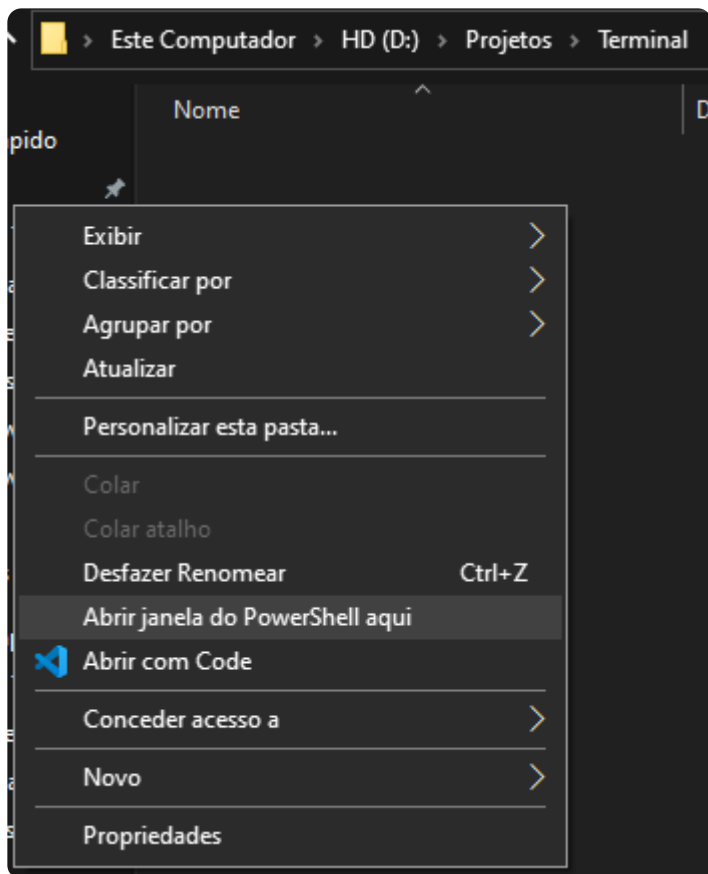
Para demonstrar melhor, vou usar o comando “**dir**”, que serve para mostrar as pastas e arquivos que estão na pasta apontada pelo terminal no Windows. Para utilizar o comando basta digitar e apertar Enter.

```
C:\Users\lucas>dir
O volume na unidade C é SSD
O Número de Série do Volume é ██████████

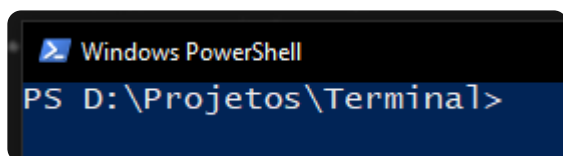
Pasta de C:\Users\lucas

06/01/2021  10:17    <DIR>        .
06/01/2021  10:17    <DIR>        ..
29/12/2020  11:34    <DIR>        .android
14/12/2020  14:07    <DIR>        .config
16/11/2020  11:36          188 .gitconfig
05/01/2021  10:42    <DIR>        .gradle
16/12/2020  11:44    <DIR>        .pylint.d
15/10/2020  16:27    <DIR>        .vscode
05/10/2020  19:52    <DIR>        3D Objects
11/10/2020  00:06    <DIR>        ansel
05/10/2020  19:52    <DIR>        Contacts
04/01/2021  18:34    <DIR>        Desktop
06/01/2021  10:17    <DIR>        Documents
06/01/2021  09:43    <DIR>        Downloads
23/10/2020  19:04          76.533 draws.bak
25/11/2020  14:54        107.520 draws.dwg
06/10/2020  11:19    <DIR>        Favorites
05/10/2020  19:52    <DIR>        Links
05/10/2020  19:52    <DIR>        Music
28/12/2020  18:37    <DIR>        Pictures
25/11/2020  14:48          685 plot.log
09/12/2020  21:25    <DIR>        Saved Games
05/10/2020  19:53    <DIR>        Searches
06/01/2021  07:01    <DIR>        Videos
           4 arquivo(s)        184.926 bytes
          20 pasta(s)    255.312.805.888 bytes disponíveis
```

Há ainda outra forma de abrir o terminal, mas dessa vez escolhendo a pasta onde ele irá apontar. Para isso, abra a pasta escolhida para iniciar o terminal no Windows, segure o SHIFT e clique com o botão DIREITO do mouse em alguma parte em branco da pasta. Atenção! Não clique em cima de nenhum arquivo ou pasta dentro daquela pasta. O resultado é o seguinte:



Dessa forma, você abrirá o PowerShell, mas como ele funciona da mesma forma não tem problema nenhum. Como resultado temos o terminal aberto já apontando para a pasta em questão para a qual navegamos.



## Navegando pelo sistema com o terminal

Vamos usar dois comandos para navegar pelas pastas do sistema: os comandos “cd” e “dir”. O “**cd**” é como um duplo clique no mouse, ou seja, ele abre a pasta, além de servir para voltar para trás.

Como ambos os terminais funcionam de forma semelhante, a partir daqui usarei apenas o PowerShell.

Com o terminal aberto na pasta selecionada, podemos usar o comando “**dir**” para descobrir o que há nela e poder avançar ou não. Para exemplificar melhor, começarei a partir da minha pasta “Projeto”, que fica no meu HD secundário.

```
Windows PowerShell
PS D:\projetos>
```

Como estamos no PowerShell, os comandos do Linux também funcionam. O comando equivalente ao “dir” no Linux é o “ls”, então também podemos usá-lo:

```
Windows PowerShell
PS D:\projetos> dir

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----          04/01/2021   10:25         alunos
d-----          27/11/2020   14:01      Atividades
d-----          01/12/2020   13:42  Projetos-Flutter-Zip
d-----          06/01/2021   10:28        Terminal

PS D:\projetos> ls

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----          04/01/2021   10:25         alunos
d-----          27/11/2020   14:01      Atividades
d-----          01/12/2020   13:42  Projetos-Flutter-Zip
d-----          06/01/2021   10:28        Terminal

PS D:\projetos> _
```

## cd + nome da pasta

A partir daqui podemos usar o comando “cd + nome da pasta” para avançar para dentro de uma pasta. Nesse caso, você deve escrever o nome correto da pasta e ela deve estar dentro da pasta atual onde você está, por exemplo:

```
Windows PowerShell
PS D:\projetos> dir

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----          04/01/2021         10:25      alunos
d-----          27/11/2020         14:01      Atividades
d-----          01/12/2020         13:42      Projetos-Flutter-Zip
d-----          06/01/2021         10:28      Terminal

PS D:\projetos> cd atividades
PS D:\projetos\atividades>
```

Dentro da nova pasta podemos utilizar o comando “dir” novamente para verificar o que há nela.

```
Windows PowerShell
PS D:\projetos> dir

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----          04/01/2021         10:25      alunos
d-----          27/11/2020         14:01      Atividades
d-----          01/12/2020         13:42      Projetos-Flutter-Zip
d-----          06/01/2021         10:28      Terminal

PS D:\projetos> cd atividades
PS D:\projetos\atividades> dir

Diretório: D:\projetos\atividades

Mode                LastWriteTime         Length Name
----                -
-a-----          19/11/2020         13:03      140761 Exercicios_de_Progra
mao_Resolucao.pdf
-a-----          27/11/2020         14:02      16144  logica.py

PS D:\projetos\atividades> _
```

**cd ..**

Podemos usar o comando “cd ..” para retornar uma pasta atrás da atual. Esse comando é sempre o mesmo, você pode usá-lo sempre que quiser ir para a pasta anterior.

```
Windows PowerShell
PS D:\projetos> dir

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----          04/01/2021    10:25             alunos
d-----          27/11/2020    14:01          Atividades
d-----          01/12/2020    13:42  Projetos-Flutter-Zip
d-----          06/01/2021    10:28           Terminal

PS D:\projetos> cd atividades
PS D:\projetos\atividades> dir

Diretório: D:\projetos\atividades

Mode                LastWriteTime         Length Name
----                -
-a----          19/11/2020    13:03    140761 Exercicios_de_Progra
mao_Resolucao.pdf
-a----          27/11/2020    14:02     16144 logica.py

PS D:\projetos\atividades> cd ..
PS D:\projetos>
```

## clear

Outro comando super interessante é o “cls” ou “clear”. Ele limpa todos os comandos listados, melhorando a visualização no terminal. Lembrando que ele limpa apenas os comandos, nenhum efeito de comando anterior é perdido.

## mkdir

Você também pode criar uma nova pasta a partir do terminal com o comando “mkdir”. No exemplo a seguir, usei o “dir” para mostrar as pastas contidas na pasta “Projetos”, criei

uma nova pasta, mostrei novamente as pastas com o “dir”, para mostrar que foi adicionada uma nova pasta lá, e entrei na pasta recém criada com o comando “cd”.

```
Windows PowerShell
PS D:\projetos> dir

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----          04/01/2021    10:25         alunos
d-----          27/11/2020    14:01      Atividades
d-----          01/12/2020    13:42  Projetos-Flutter-Zip
d-----          06/01/2021    10:28        Terminal

PS D:\projetos> mkdir testando

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----          06/01/2021    10:46        testando

PS D:\projetos> dir

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----          04/01/2021    10:25         alunos
d-----          27/11/2020    14:01      Atividades
d-----          01/12/2020    13:42  Projetos-Flutter-Zip
d-----          06/01/2021    10:28        Terminal
d-----          06/01/2021    10:46        testando

PS D:\projetos> cd testando
PS D:\projetos\testando> 
```

## del + nome da pasta/arquivo

Você pode também apagar arquivos e pastas com o comando “del + nome da pasta/arquivo”. Da seguinte forma:



```
Windows PowerShell
PS D:\projetos> dir

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----         04/01/2021         10:25      alunos
d-----         27/11/2020         14:01      Atividades
d-----         01/12/2020         13:42      Projetos-Flutter-Zip
d-----         06/01/2021         10:28      Terminal
d-----         06/01/2021         10:50      testando

PS D:\projetos> del testando
PS D:\projetos> dir

Diretório: D:\projetos

Mode                LastWriteTime         Length Name
----                -
d-----         04/01/2021         10:25      alunos
d-----         27/11/2020         14:01      Atividades
d-----         01/12/2020         13:42      Projetos-Flutter-Zip
d-----         06/01/2021         10:28      Terminal

PS D:\projetos>
```

## Help

Um outro comando super válido de se utilizar é o 'help', pois ele lista todos os comandos que você pode utilizar. É muito bom para quem não quer ir pesquisar em outro local sobre os demais comandos do terminal. Basta utilizar o comando 'help' dentro dele mesmo e pronto!

```
C:\Users\lucas>help
Para obter mais informações sobre um comando específico,
digite HELP nome_do_comando
ASSOC           Exibe ou modifica associações de extensões de arquivo.
ATTRIB          Exibe ou altera atributos de arquivos.
BREAK           Define ou limpa a verificação estendida CTRL+C.
BCDEDIT          Define propriedades no banco de dados de inicialização para
                controlar o carregamento da inicialização.
CACLS           Exibe ou modifica listas de controle de acesso de arquivos.
CALL            Chama um programa em lotes por meio de outro.
CD              Exibe o nome do diretório atual ou faz alterações nele.
CHCP            Exibe ou define o número da página de código ativa.
CHDIR           Exibe o nome do diretório atual ou faz alterações nele.
CHKDSK          Verifica um disco e exibe um relatório de status.
CHKNTFS         Exibe ou modifica a verificação do disco na inicialização.
CLS             Limpa a tela.
CMD             Inicia uma nova instância do interpretador de comandos do
                Windows.
COLOR           Define as cores padrão do primeiro plano e da tela de fundo
                do console.
COMP            Compara o conteúdo de dois arquivos ou grupos de arquivos.
COMPACT         Exibe ou altera a compactação de arquivos em partições NTFS.
CONVERT         Converte volumes FAT em NTFS. Não é possível converter a
                unidade atual.
COPY            Copia um ou mais arquivos para outro local.
DATE           Exibe ou ajusta a data.
DEL            Exclui um ou mais arquivos.
DIR            Exibe uma lista de arquivos e subdiretórios em um diretório.
DISKPART        Exibe ou configura propriedades de partição de disco.
```

## Outros usos

Várias tecnologias quando instaladas na sua máquina têm o potencial de serem utilizadas pelo terminal. Um exemplo disso é o **NodeJS**, uma vez instalada na sua máquina você pode utilizar comandos com o 'npm' ou 'node', que indicam que você estará utilizando aquele software.

```
C:\Users\lucas>node --version
v14.15.1

C:\Users\lucas>npm --version
7.0.15

C:\Users\lucas>_
```

Lembrando que no caso do Node é instalado um terminal próprio do Node que recebe comandos JavaScript e esse terminal do Node NÃO funciona como o CMD e o PowerShell do Windows, então não abram ele e tentem utilizar pois isso irá gerar erros nos comandos.

Outro caso é o **Flutter e o Dart**, que também se utilizam do terminal após instalados na máquina, sendo capazes de criar projetos, se atualizar dentre outras coisas.

Esses outro comandos variam de tecnologia para tecnologia, de versão para versão, então não irei dar tantos exemplos aqui, mas sempre que se instalar uma nova tecnologia é bom lembrar de verificar e aprender quais comandos aquela tecnologia possui no seu terminal.

Esse pequeno guia é apenas um compilado de pequenas dicas para pessoas que tiveram pouco ou nenhum contato com o terminal consigam lidar de forma mais efetiva com essa ferramenta tão importante. E que também tenham menos medo dele! Espero ter ajudado. Bons estudos e até a próxima!