

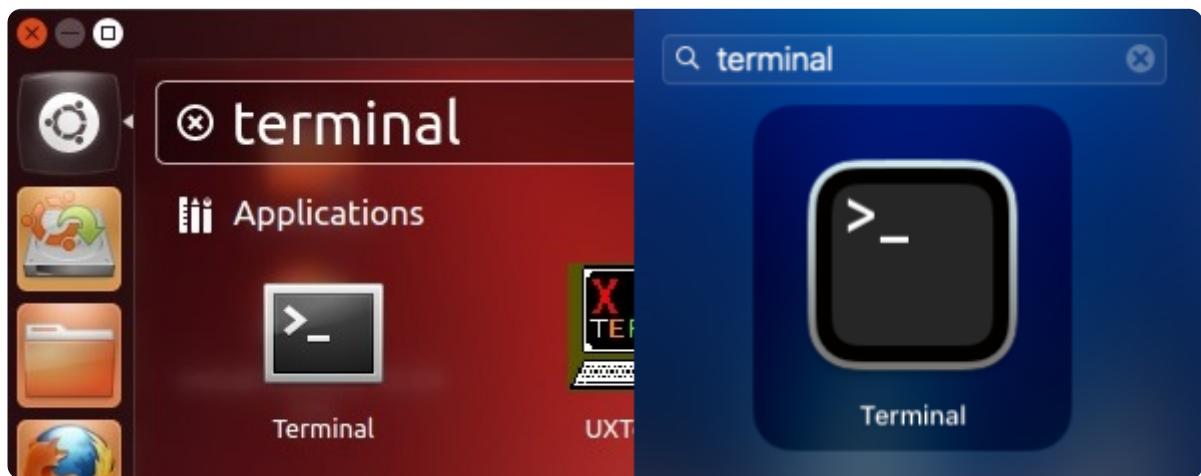
Terminal no MacOS e Linux

O **Terminal** no MacOS e Linux é uma ferramenta indispensável para os profissionais da TI, desde o programador até o profissional de redes. Afinal, é uma ferramenta capaz de fazer instalações, configurações, checar dados e navegar pelo sistema.

Mesmo não sendo muito conhecido no mundo da programação, não são todos que gostam ou sabem usá-lo. Se esse é seu caso, vamos te ajudar nesse artigo!

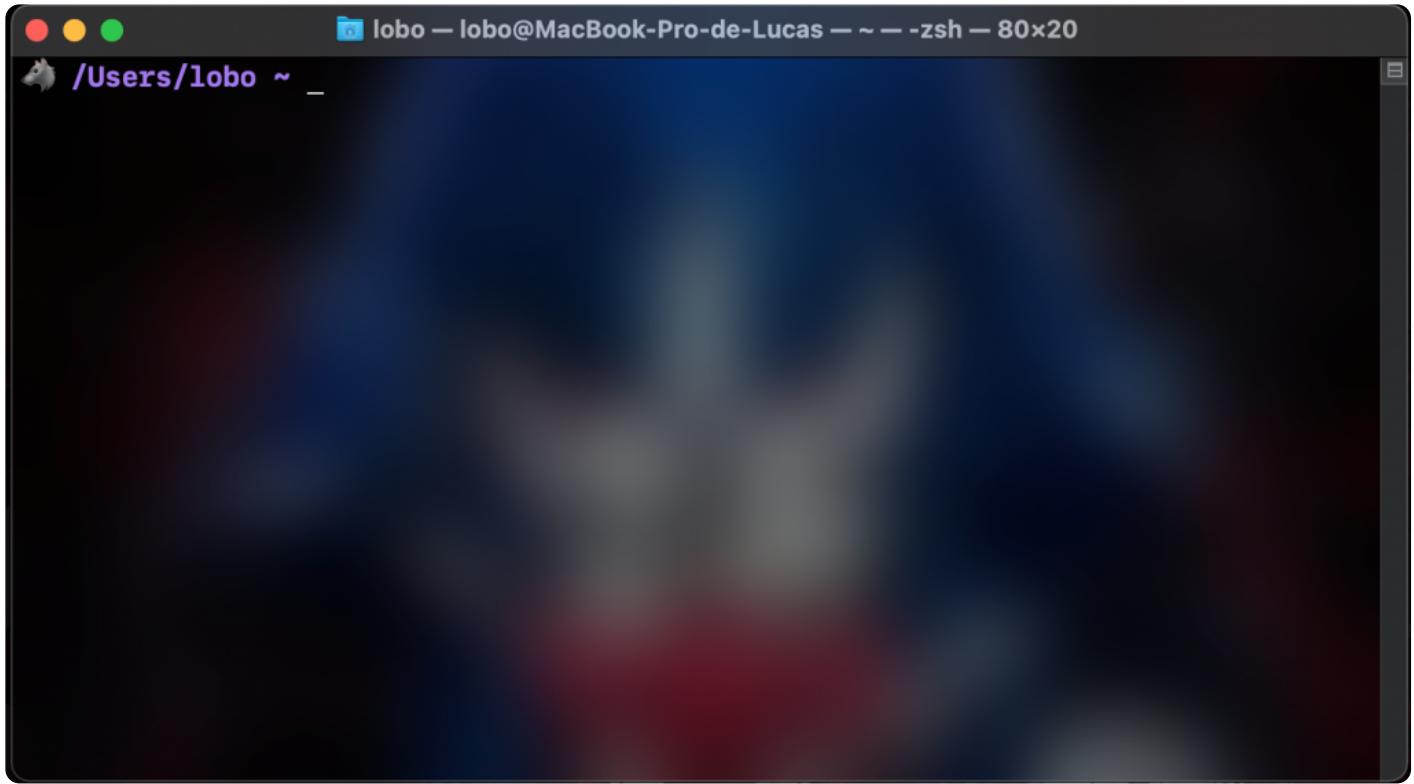
Conhecendo o Terminal

Para abrir o Terminal no MacOS e Linux procure por “Terminal” na sua lista de Programas/Aplicativos. No Mac, você pode usar o Spotlight com o atalho “COMMAND + ESPAÇO” e digitar “terminal” para abri-lo.

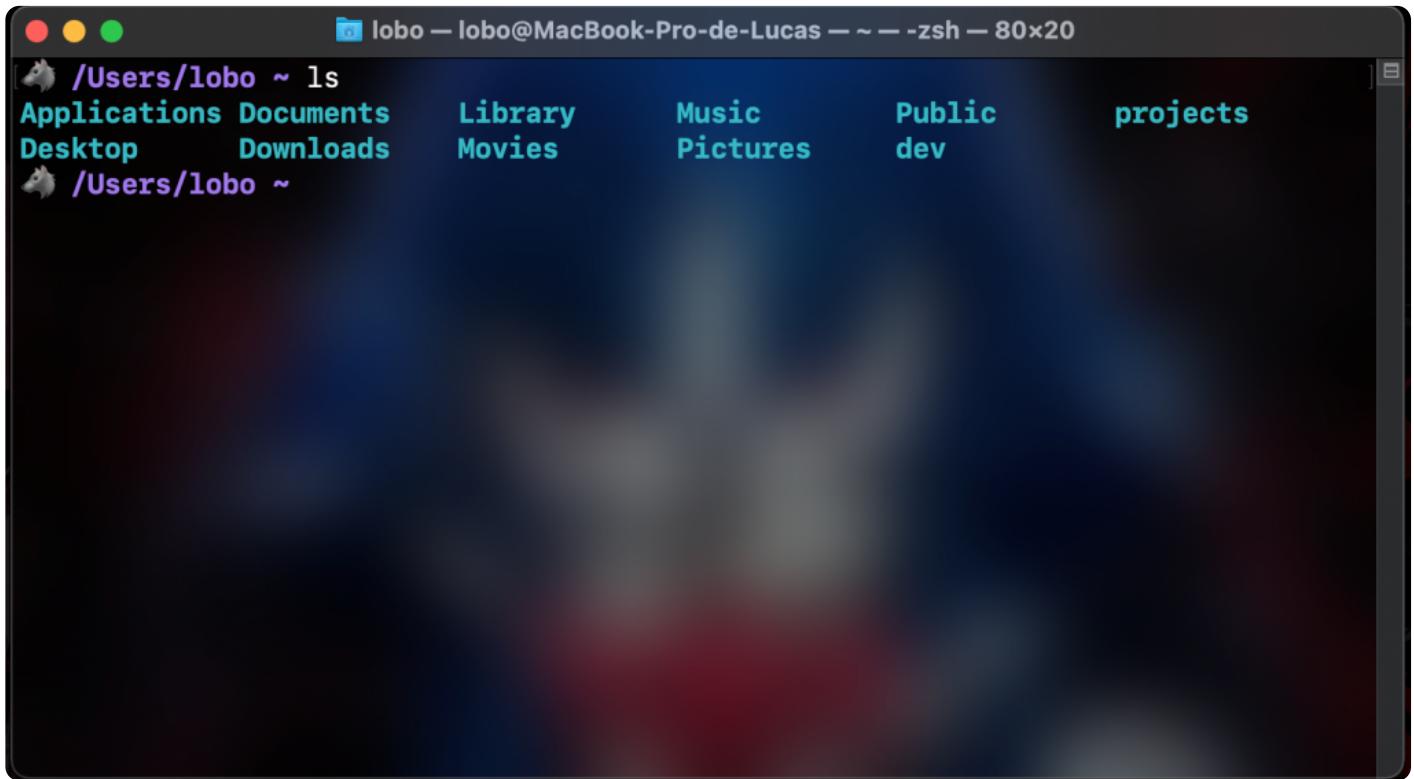


Abrindo dessa forma, ele apontará diretamente para a pasta do seu usuário, ou seja, para o caminho: C, que é o HD principal da sua máquina, onde o seu OS está instalado; e Users, que é a pasta de usuários e seu nome de usuário.

Pode ser que o que apareça na sua tela varie um pouco de acordo com seu sistema e a versão dele. Mas o padrão é o seguinte:



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



Você também pode escolher qual pasta será apontada ao abrir o Terminal. Basta abrir o explorador de Arquivos (no Mac é o Finder) e navegue até a pasta que você quer abra no terminal. Em seguida, clique com o botão direito na pasta e selecione a opção de abrir um novo Terminal na pasta. No Mac você verá o seguinte:

Abrir em Nova Aba

Mover para o Lixo

Obter Informações

Renomear

Comprimir "github"

Duplicar

Criar Atalho

Visualização Rápida

Copiar

Compartilhar >



Etiquetas...

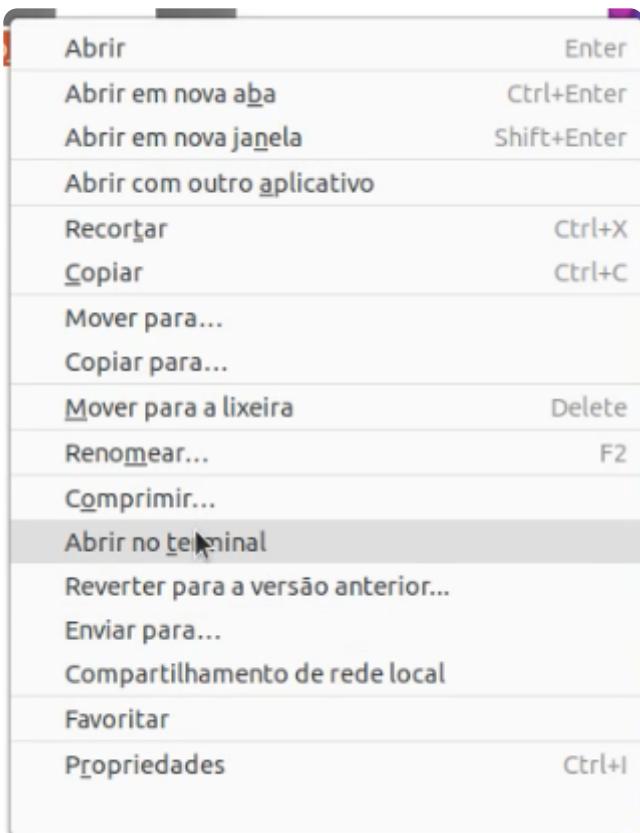
Ações Rápidas >

Configuração de Ações de Pasta...

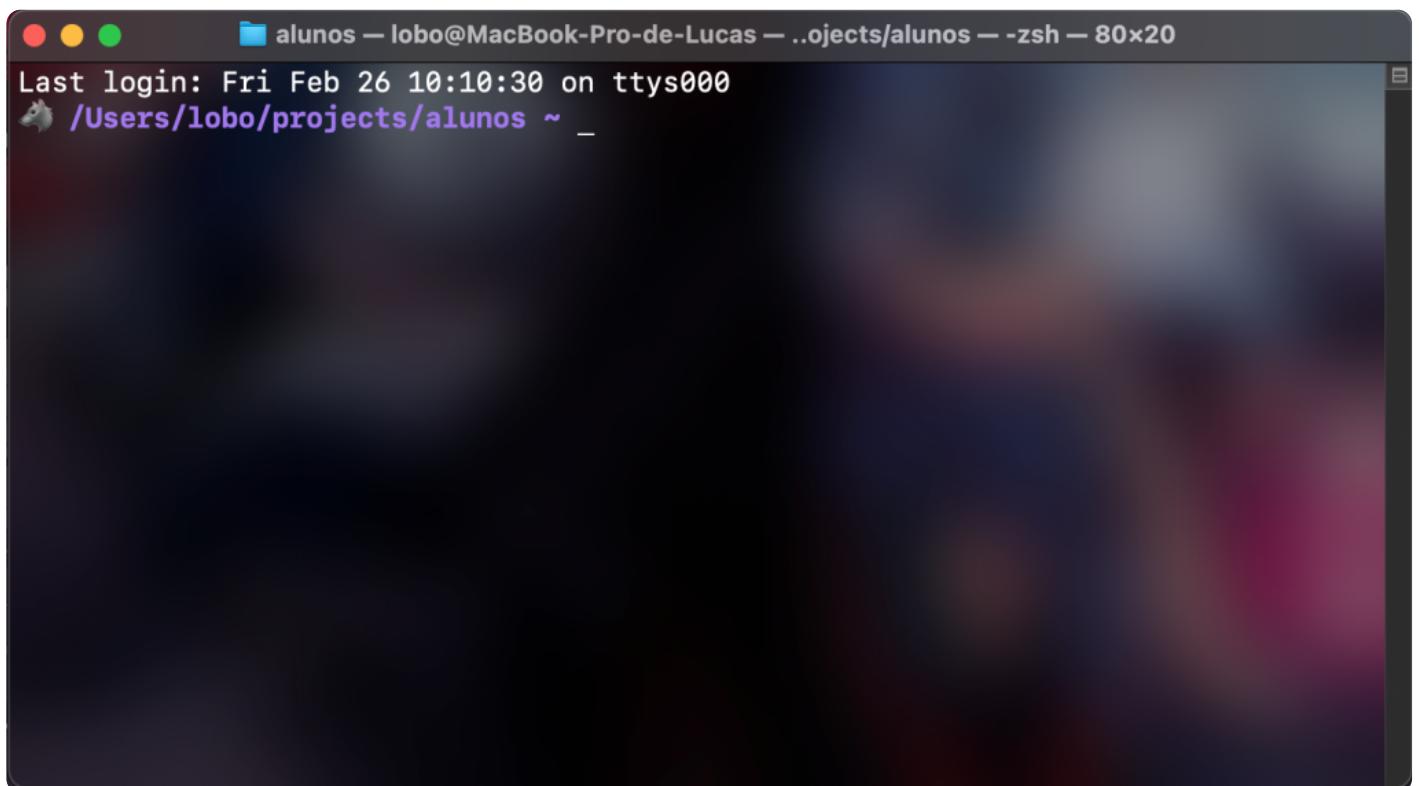
Nova Aba de Terminal na Pasta

Novo Terminal na Pasta

E no Ubuntu, assim:

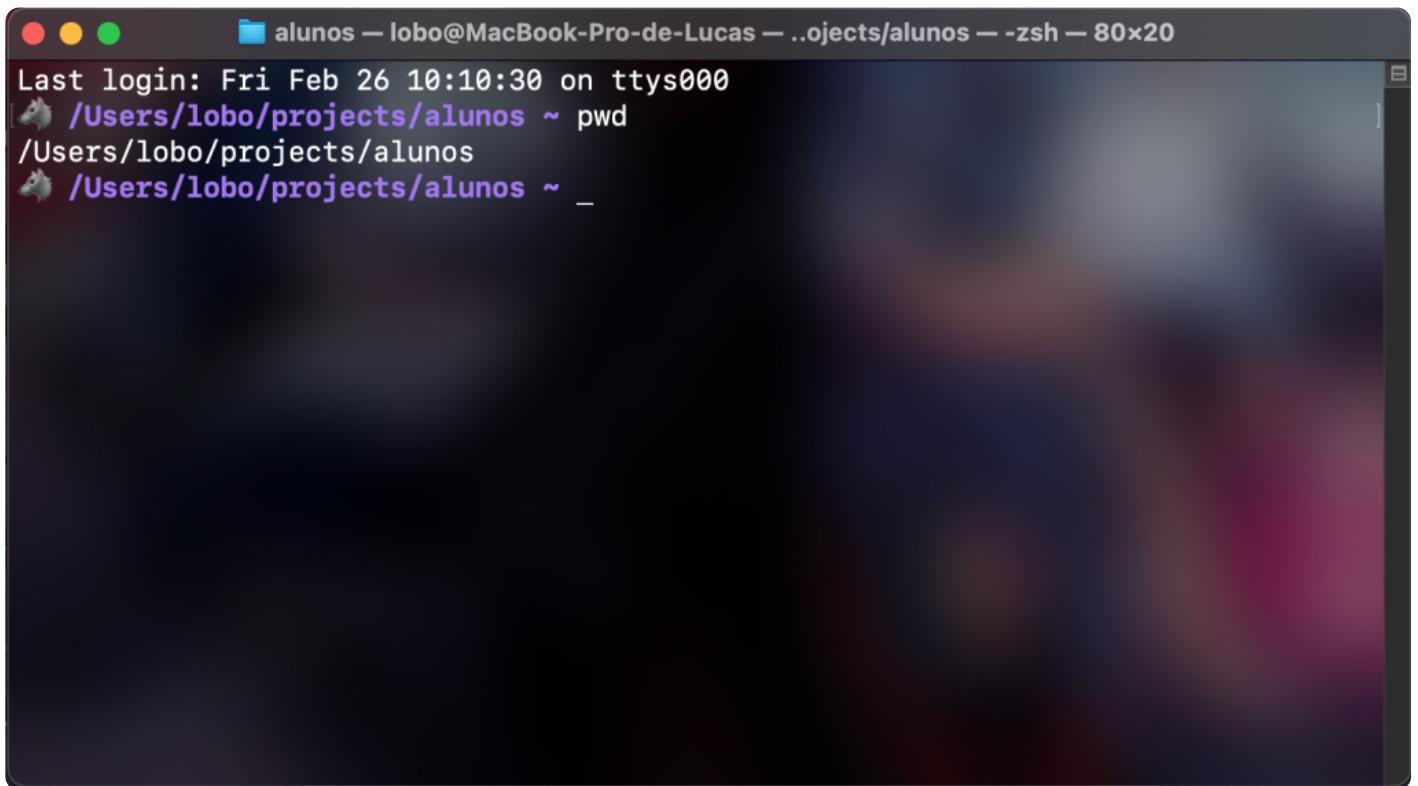


Ao clicar nessa opção, o Terminal abrirá diretamente naquela pasta.



Se o caminho para a pasta não aparecer no terminal, você pode testar se deu tudo certo com o comando "pwd". Esse comando printa o caminho atual no terminal, ou seja, para

onde ele está apontando.

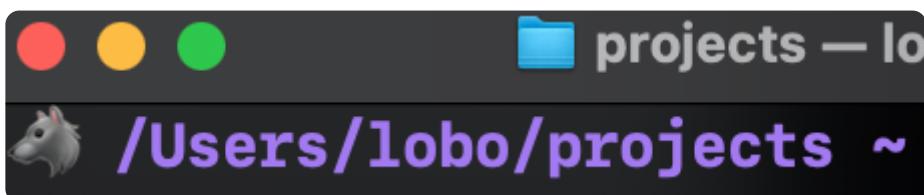


```
alunos — lobo@MacBook-Pro-de-Lucas — ..ojects/alunos — -zsh — 80x20
Last login: Fri Feb 26 10:10:30 on ttys000
[lobo: ~] /Users/lobo/projects/alunos ~ pwd
/Users/lobo/projects/alunos
[lobo: ~]
```

Navegando pelo sistema com o terminal

Vamos usar dois comandos para navegar pelas pastas do sistema: os comandos “cd” e “ls”. O “cd” funciona como um duplo clique no mouse, ou seja, ele abre a pasta e retorna para a pasta anterior.

O “ls”, quando o terminal está aberto na pasta selecionada, serve para descobrir o que há nela e avançar ou não. Para exemplificar melhor, começarei a partir da minha pasta “projects”, que fica no meu HD principal.



```
projects — lo
[lobo: ~] /Users/lobo/projects ~
```

Primeiramente, vamos usar o comando “ls” para saber quais pastas estão dentro da pasta atual.

```
● ● ● projects — lobo@iMac:~/Projects
```

```
horse /Users/lobo/projects ~ ls
alunos github navbar
horse /Users/lobo/projects ~ _
```

Você também pode usar algumas variações do comando ls, como por exemplo a flag “a” que mostra todos os arquivos, inclusive os escondidos.

```
● ● ● projects — lobo@MacBook-Pro-de-Lucas — ~/Projects —
horse /Users/lobo/projects ~ ls -a
. .. .DS_Store alunos github navbar
horse /Users/lobo/projects ~
```

Outra flag é “l”, que mostra a versão longa do “ls” normal, ou seja, traz mais informações sobre os arquivos.

```
● ● ● projects — lobo@MacBook-Pro-de-Lucas — ~/Projects —
horse /Users/lobo/projects ~ ls -l
total 0
drwxr-xr-x 5 lobo staff 160 28 Jan 17:09 alunos
drwxr-xr-x 4 lobo staff 128 18 Fev 23:16 github
drwxr-xr-x 15 lobo staff 480 16 Fev 17:37 navbar
horse /Users/lobo/projects ~ _
```

Você também pode juntar as flags e usá-las ao mesmo tempo:

```
projects — lobo@MacBook-Pro-de-Lucas — ~/projects —  
/Users/lobo/projects ~ ls -al  
total 24  
drwxr-xr-x@ 6 lobo staff 192 17 Fev 16:39 .  
drwxr-xr-x+ 34 lobo staff 1088 26 Fev 10:53 ..  
-rw-r--r--@ 1 lobo staff 8196 26 Fev 10:38 .DS_Store  
drwxr-xr-x 5 lobo staff 160 28 Jan 17:09 alunos  
drwxr-xr-x 4 lobo staff 128 18 Fev 23:16 github  
drwxr-xr-x 15 lobo staff 480 16 Fev 17:37 navbar  
/Users/lobo/projects ~
```

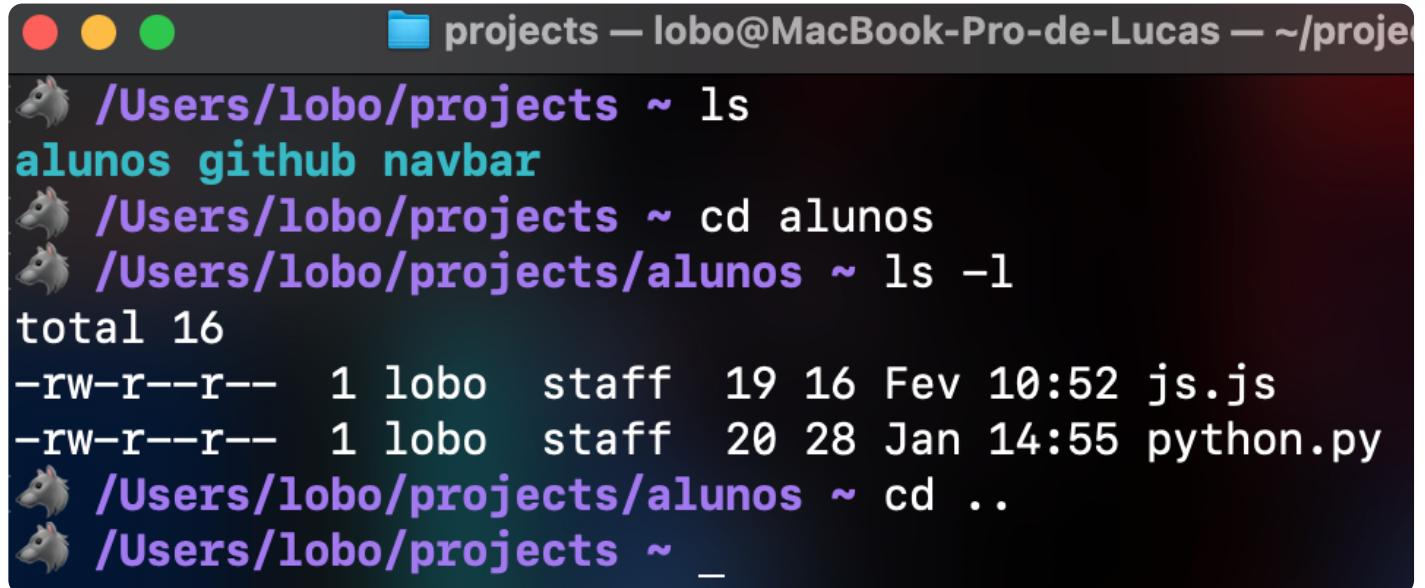
De agora em diante, podemos usar o comando “cd + nome da pasta” para avançar para dentro de uma pasta. Nesse caso, você deve escrever o nome correto da pasta e ela deve estar dentro da pasta atual onde você está, por exemplo:

```
alunos — lobo@MacBook-Pro-  
/Users/lobo/projects ~ ls  
alunos github navbar  
/Users/lobo/projects ~ cd alunos  
/Users/lobo/projects/alunos ~ _
```

Dentro da nova pasta podemos usar o comando “ls” para verificar seu conteúdo:

```
alunos — lobo@MacBook-Pro-de-Lucas — ..ojects/al  
/Users/lobo/projects ~ ls  
alunos github navbar  
/Users/lobo/projects ~ cd alunos  
/Users/lobo/projects/alunos ~ ls -l  
total 16  
-rw-r--r-- 1 lobo staff 19 16 Fev 10:52 js.js  
-rw-r--r-- 1 lobo staff 20 28 Jan 14:55 python.py  
/Users/lobo/projects/alunos ~ _
```

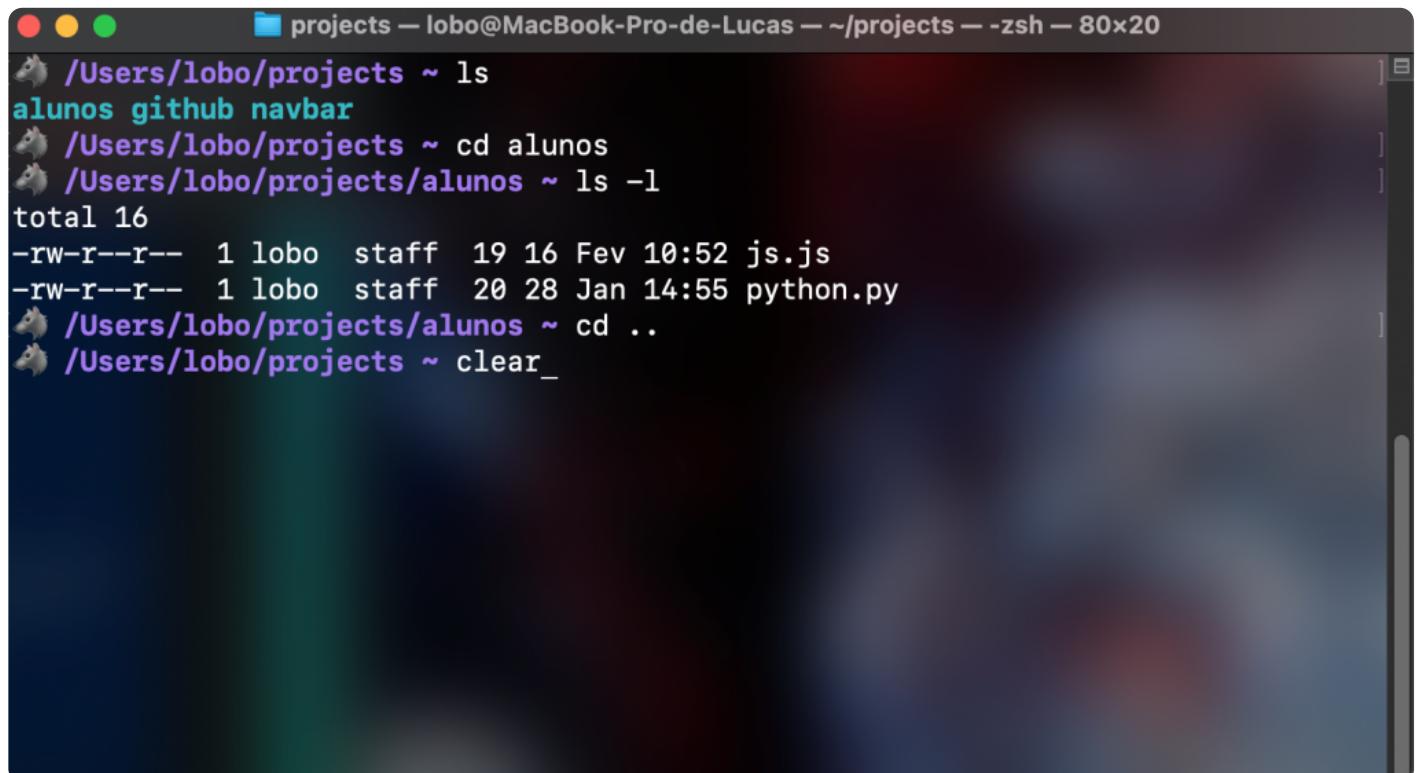
Podemos usar também o comando “cd ..” para retornar para a pasta anterior.



```
projects — lobo@MacBook-Pro-de-Lucas — ~/proje
/Users/lobo/projects ~ ls
alunos github navbar
/Users/lobo/projects ~ cd alunos
/Users/lobo/projects/alunos ~ ls -l
total 16
-rw-r--r-- 1 lobo  staff  19 16 Fev 10:52 js.js
-rw-r--r-- 1 lobo  staff  20 28 Jan 14:55 python.py
/Users/lobo/projects/alunos ~ cd ..
/Users/lobo/projects ~ _
```

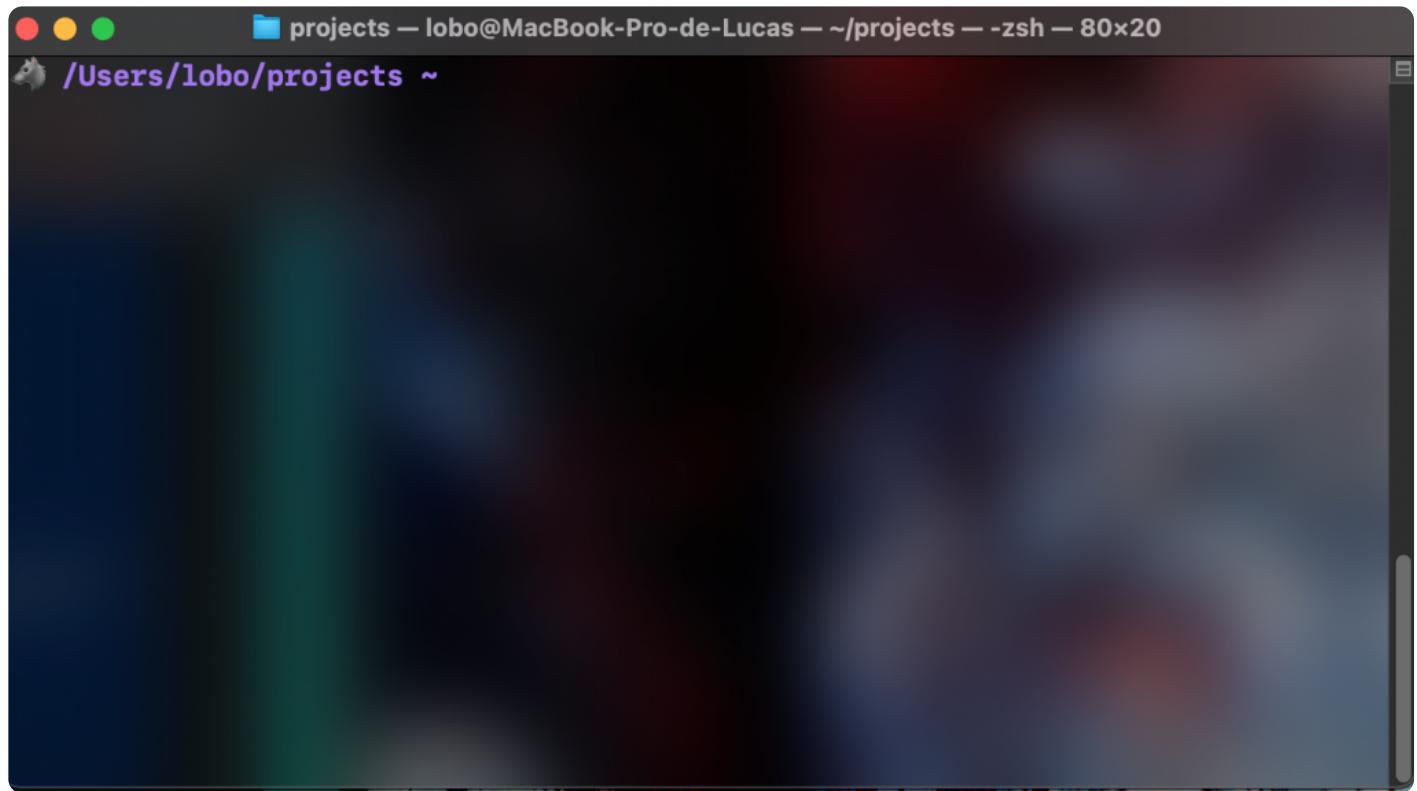
Limpando o Terminal

Após usarmos vários comandos, um após o outro, o Terminal acaba ficando poluído.



```
projects — lobo@MacBook-Pro-de-Lucas — ~/projects — zsh — 80x20
/Users/lobo/projects ~ ls
alunos github navbar
/Users/lobo/projects ~ cd alunos
/Users/lobo/projects/alunos ~ ls -l
total 16
-rw-r--r-- 1 lobo  staff  19 16 Fev 10:52 js.js
-rw-r--r-- 1 lobo  staff  20 28 Jan 14:55 python.py
/Users/lobo/projects/alunos ~ cd ..
/Users/lobo/projects ~ clear_
```

Por isso usamos o comando “clear” para limpar a tela.



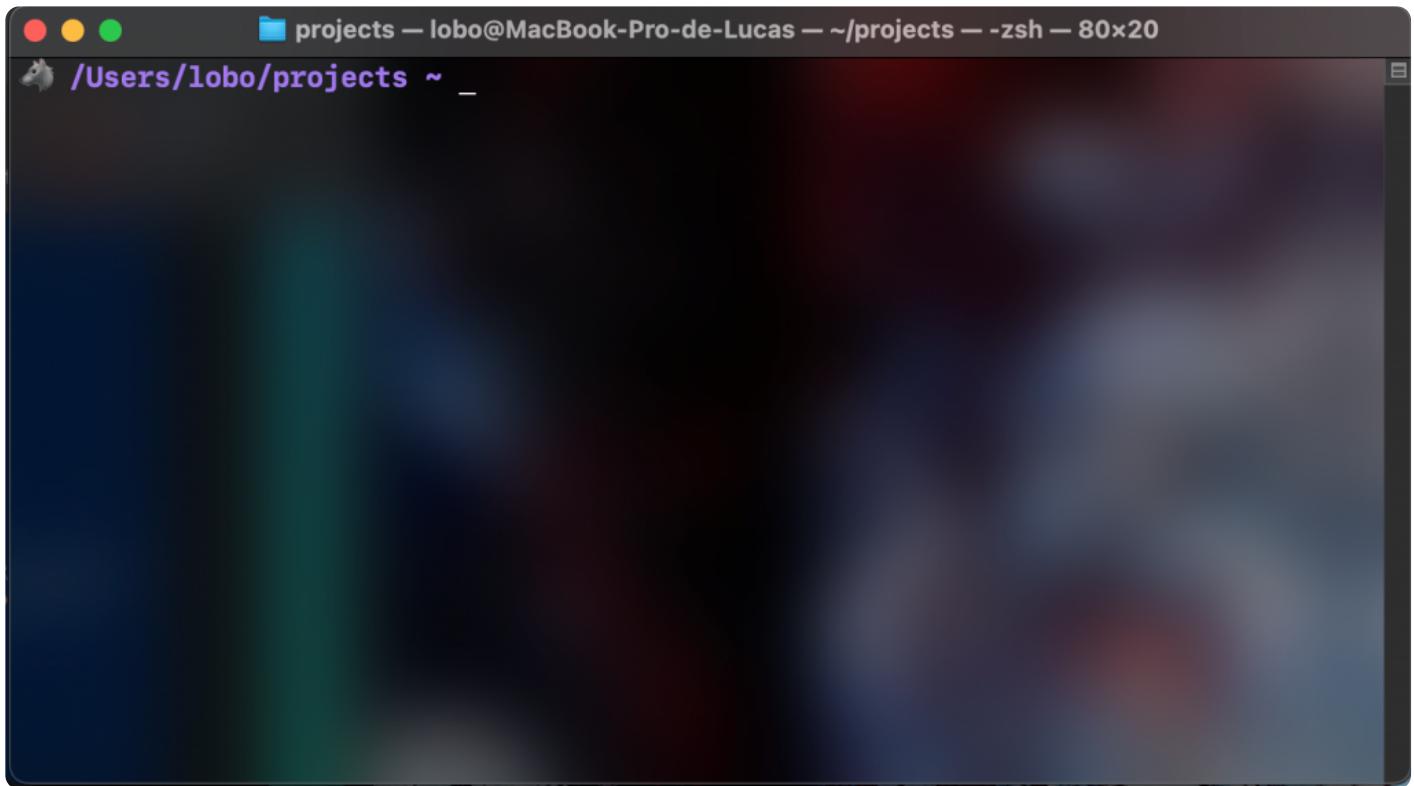
Um fato interessante é que no MacOS o comando “clear” não limpa de fato o terminal, mas apenas printa várias linhas vazias para que os comandos usados anteriormente subam na tela e não fiquem visiveis.

```
projects — lobo@MacBook-Pro-de-Lucas — ~/projects — zsh — 80x42

[1] 100% /Users/lobo/projects ~ ls
alunos github navbar
[2] 100% /Users/lobo/projects ~ cd alunos
[3] 100% /Users/lobo/projects/alunos ~ ls -l
total 16
-rw-r--r-- 1 lobo  staff  19 16 Fev 10:52 js.js
-rw-r--r-- 1 lobo  staff  20 28 Jan 14:55 python.py
[4] 100% /Users/lobo/projects/alunos ~ cd ..
[5] 100% /Users/lobo/projects ~ clear

[6] 100% /Users/lobo/projects ~ _
```

Para limpar o terminal de verdade, você deve usar o atalho “COMMAND + K”. Dessa forma o terminal será de fato zerado e renovado.



Criando pastas via terminal

Você também pode criar uma nova pasta a partir do terminal com o comando “mkdir”.

No exemplo abaixo eu usei o “ls” para mostrar as pastas contidas dentro na pasta “projects”; criei uma nova pasta; mostrei novamente as pastas com o “ls”, para mostrar que foi adicionada uma nova pasta lá; e entrei na pasta recém criada com o comando “cd”.



teste — lobo@MacBook-Pro-de-

```
[`/Users/lobo/projects ~ ls
alunos github navbar
[`/Users/lobo/projects ~ mkdir teste
[`/Users/lobo/projects ~ ls
alunos github navbar teste
[`/Users/lobo/projects ~ cd teste
[`/Users/lobo/projects/teste ~
```

Deletando pastas e arquivos via terminal

Para deletar arquivos pelo terminal basta usar o comando “rm” para deletar arquivos, e “rmdir” para deletar pastas. Vou exemplificar deletando a pasta que criei no exemplo anterior.



projects — lobo@MacBook-Pro

```
[`/Users/lobo/projects ~ ls
alunos github navbar teste
[`/Users/lobo/projects ~ rmdir teste
[`/Users/lobo/projects ~ ls
alunos github navbar
[`/Users/lobo/projects ~
```

Outros usos

Várias tecnologias instaladas na sua máquina podem ser usadas pelo terminal, como por exemplo o NodeJS, Flutter e Dart. Após instalar, você pode usar os comandos com o ‘npm’, ‘node’, ‘dart’ ou ‘flutter’ para indicar que você quer usar aquele software especificamente.

```
projects — lobo@MacBook-Pro-de-Lucas — ~/projects — -zsh — 80x20
[lobo:~/projects ~] $ npm --version
6.14.10
[lobo:~/projects ~] $ node --version
v14.15.4
[lobo:~/projects ~] $ flutter --version
Flutter 1.22.6 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 9b2d32b605 (5 weeks ago) • 2021-01-22 14:36:39 -0800
Engine • revision 2f0af37152
Tools • Dart 2.10.5
[lobo:~/projects ~] $ dart --version
Dart SDK version: 2.10.5 (stable) (Tue Jan 19 13:05:37 2021 +0100) on "macos_x64"
"
[lobo:~/projects ~] $
```

Esses outros comandos variam de tecnologia para tecnologia, de versão para versão, então não irei dar tantos exemplos aqui. Mas sempre que você instalar uma nova tecnologia, é bom verificar quais comandos podem ser usados no terminal.

Nesses dois sites existem alguns links com vários comandos e dicas que você pode usar no terminal:

<https://www.makeuseof.com/tag/mac-terminal-commands-cheat-sheet/> <https://www.guru99.com/linux-commands-cheat-sheet.html>

Esse pequeno guia é apenas um compilado de pequenas dicas para pessoas que tiveram pouco ou nenhum contato com o terminal no MacOS e Linux consigam lidar de forma mais efetiva com essa ferramenta tão importante. E que também tenham menos medo dele! Espero ter ajudado.