



Aprenda a programar

Dia 1 - Primeiros passos e preparando a caixa de ferramentas

O que veremos

Alinhando as expectativas

Como o curso vai funcionar

O que é programação?

Qualquer pessoa pode ser programadora?

Preparando a caixa de ferramentas

Introdução ao terminal

O poder do GitHub para a
sua carreira em programação

Alinhando as expectativas

Alinhando as expectativas

O que queremos com este curso?



Veja se programação é ou não é para você



Plantar a semente para que você adquira o hábito de programar



Saiba de fato o que é programação e como escrever comandos para o computador



Tenha clareza sobre os próximos passos para uma carreira em programação



Entenda o mundo de oportunidades que existe na área de tecnologia

Alinhando as expectativas

O que NÃO É o objetivo deste curso?



Seja **expert** em programação após terminar o curso

Como o curso vai funcionar

Como o curso vai funcionar



5 dias de muita mão na massa



Tudo vai acontecer no Discord!



Certificado ao final do curso

Como o curso vai funcionar

31

5 dias de muita mão na massa

Dia 1 (Segunda - 22/08): Primeiros passos e preparando a caixa de ferramentas

Dia 2 (Terça - 23/08): Aprendendo a programar e adquirindo o hábito com GitHub, parte 1

Dia 3 (Quarta - 24/08): Aprendendo a programar e adquirindo o hábito com GitHub, parte 2

Dia 4 (Quinta - 25/08): Conhecendo o mundo da web com HTML + CSS + JavaScript

Dia 5 (Sexta - 26/08): Seus próximos passos como pessoa programadora

Como o curso vai funcionar



Tudo vai acontecer no Discord!

- Todas as tarefas serão adicionadas no Discord
- O material das aulas serão disponibilizados lá
- Agenda das aulas e prazo das tarefas também estarão lá
- Interação, bate-papo e tira-dúvidas

Como o curso vai funcionar



Certificado ao final do curso



Como obter o certificado?

- Enviar todas as tarefas do curso nos formulários Indicados **no prazo informado**
- Responder o formulário de avaliação de cada um dos dias do curso

O que é programação?

O que é programação?



Resolver problemas utilizando
instruções que podem ser executadas
por um computador

O que é programação?



O que é programação?

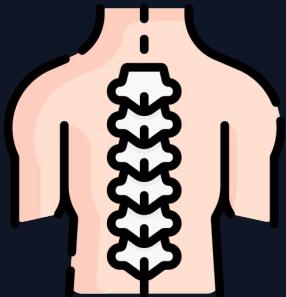


Programação é a profissão que mais cresce no mundo.

No LinkedIn, hoje, existem cerca de **60000** vagas abertas para programadores no Brasil. Segundo a Forbes*, a estimativa é que até 2030, faltarão 85 milhões de profissionais de TI no mundo inteiro.

*<https://www.forbes.com/sites/forbestechcouncil/2021/06/08/is-there-a-developer-shortage-yes-but-the-problem-is-more-complicated-than-it-looks/?sh=442lc8ac3b8e>

O que é programação?



Programação é a **espinha dorsal** de praticamente qualquer produto, serviço ou empresa de tecnologia.

O que é programação?



O que é programação?

Os vários papéis em programação...



Engenharia ou ciência de dados



Desenvolvimento Backend



Engenharia de
Inteligência Artificial



Desenvolvimento
Mobile



Desenvolvimento
Front-end

Tudo se resume a programação,
porém com o foco em resolver
problemas distintos

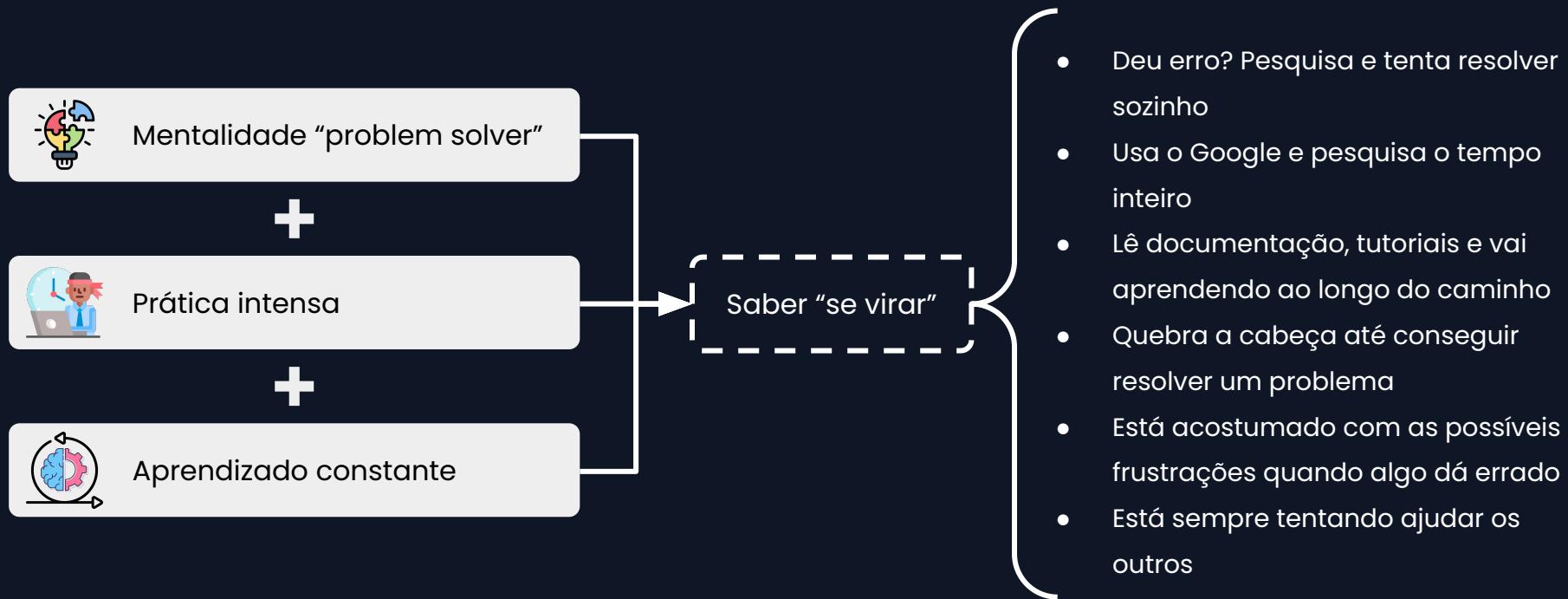


Engenharia DevOps

Qualquer pessoa pode ser programadora?

Qualquer pessoa pode ser programadora?

Existe uma forma de pensar e agir que é comum para quem desenvolve software?



Qualquer pessoa pode ser programadora?

Existe uma forma de pensar e agir que é comum para quem desenvolve software?

Saber “se virar” é algo que se aprende.

Qualquer pessoa pode ser programadora?

Existe uma forma de pensar e agir que é comum para quem desenvolve software?



Devo amar programação para
trabalhar com ela?

Qualquer pessoa pode ser programadora?

A filosofia por trás da programação de computadores



Codifique



Aplique,
aprenda, erre e
conserte!



Evolua!

Saiba que em programação você vai errar o tempo inteiro!

Qualquer pessoa pode ser programadora?

A filosofia por trás das empresas de tecnologia



Faça



Aplique,
aprenda, erre e
conserte!



Evolua!

**A maioria das empresas de tecnologia no mundo seguem ou
buscam seguir uma cultura de melhoria contínua e de agilidade**

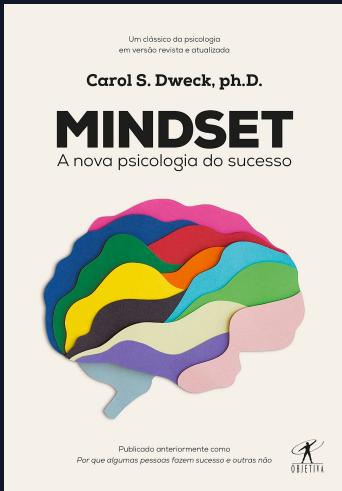
Qualquer pessoa pode ser programadora?

E será que sou realmente capaz?

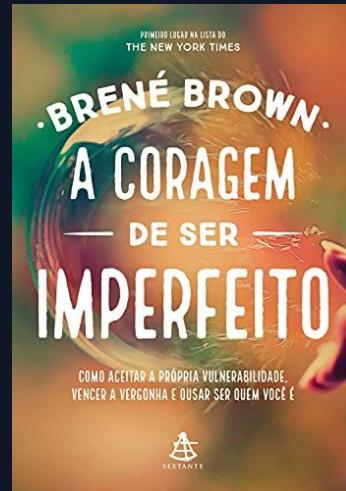


Qualquer pessoa pode ser programadora?

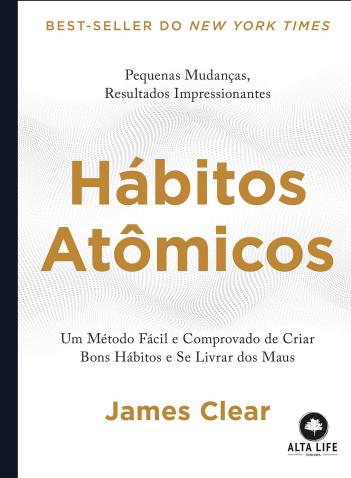
E será que sou realmente capaz?



[Vídeo sobre mentalidade de crescimento](#)



[Vídeo sobre o poder da vulnerabilidade](#)



[Vídeo sobre hábitos](#)

Qualquer pessoa pode ser programadora?

**Utilize o nosso ebook como guia
ao longo da sua jornada com a
gente! :)**



Preparando a caixa de ferramentas que toda pessoa programadora utiliza

Preparando a caixa de ferramentas de programação



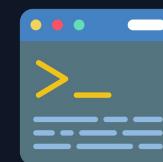
E dessas ferramentas, quais vamos utilizar aqui no curso?



Editor de texto VS CODE
(não é Microsoft Word e nem Google Docs!!!)



Veja e controle o histórico do que você está codificando com o git



Um terminal para se comunicar com o sistema operacional e executar o programa que você codificou

Linguagens de programação* que serão utilizadas no curso:



JavaScript



HTML



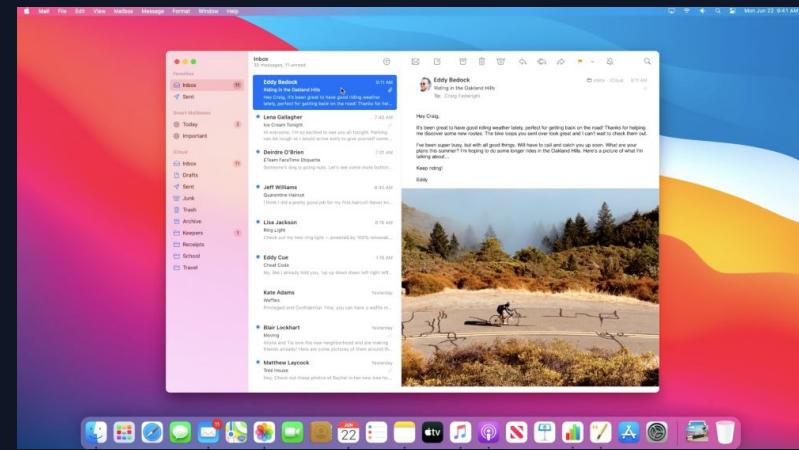
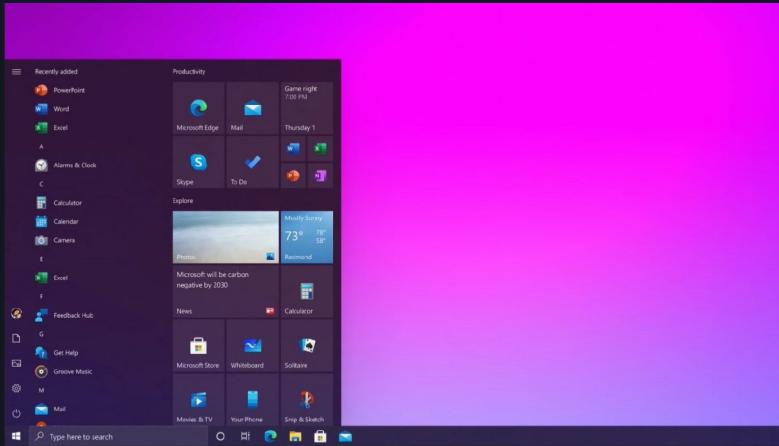
CSS

*html e css não são linguagens de programação. São linguagens com um outro propósito que vamos explicar mais pra frente

Introdução ao terminal

Introdução ao terminal

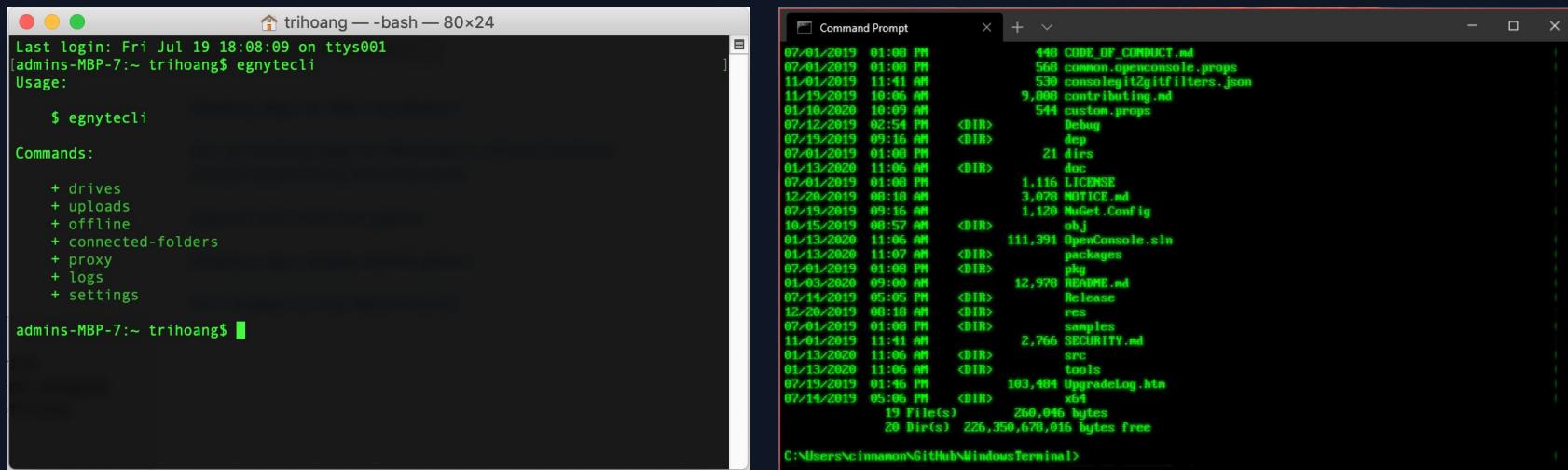
Pessoas comuns interagem com o sistema operacional por meio da interface de usuário.



Introdução ao terminal

Em programação, vamos precisar muitas vezes interagir com o sistema operacional por meio de comandos de texto para (principalmente):

- Executar o código que escrevemos
- Executar comandos para configurar pastas, sistemas e ferramentas



The image displays two side-by-side terminal windows. The left window is a macOS terminal window titled 'trihoang — bash — 80x24'. It shows a login message, a usage statement for 'egnytecli', and a list of commands: '+ drives', '+ uploads', '+ offline', '+ connected-folders', '+ proxy', '+ logs', and '+ settings'. The right window is a Windows Command Prompt window titled 'Command Prompt'. It shows a detailed file listing from a GitHub repository, including files like 'CODE_OF_CONDUCT.md', 'common.openconsole.props', 'consolegit2gitfilters.json', 'contributing.md', 'custom.props', 'Debug', 'dep', 'dirs', 'doc', 'LICENSE', 'NOTICE.md', 'NuGet.Config', 'obj', 'OpenConsole.sln', 'packages', 'pkg', 'README.md', 'Release', 'res', 'samples', 'SECURITY.md', 'src', 'tools', 'UpgradeLog.htm', and 'x64'. The total file count is 19 and the total size is 260,046 bytes.

O propósito do terminal é este.

Introdução ao terminal

Pra não se confundir com o terminal, o importante é:



```
Last login: Fri Jul 19 18:08:09 on ttys01
[admins-MBP-7:~ trihoang$ egnytecli
Usage:
$ egnytecli
Commands:
+ drives
+ uploads
+ offline
+ connected-folders
+ proxy
+ logs
+ settings
admins-MBP-7:~ trihoang$
```

Utilize o comando **pwd** para sempre saber em que pasta você está.

No terminal, você **sempre** está em algum lugar!

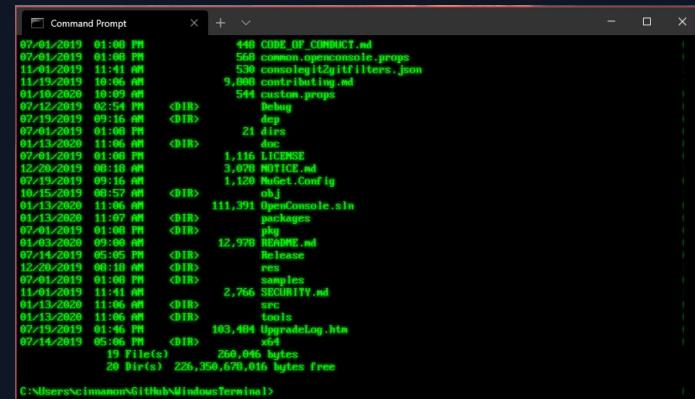
O terminal é como um navegador de pastas. Você sempre estará em alguma pasta do seu computador.

Por isso, você precisa sempre saber em que pasta está para fazer as coisas corretamente.

Introdução ao terminal

Hora da prática!

- **Principais comandos nativos do terminal:**
 - pwd - mostrar em que pasta estamos
 - ls - listar tudo que está dentro de uma pasta
 - cd - mudar de diretório
 - mkdir - criar uma pasta/diretório
 - touch - criar um arquivo
 - cat - mostrar todo o conteúdo do arquivo
 - head - mostrar o que está no início do arquivo
 - grep - pesquisar conteúdo em arquivos



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window displays a list of files and directories from a folder. The output is as follows:

```
07-01-2019 01:00 PM           448 CODE_OF_CONDUCT.md
07-01-2019 01:00 PM           568 common_openconsole.props
11-01-2019 11:41 AM          530 consolelegitgitfilters.json
11-01-2019 01:00 PM          9,888 contributing.md
01-10-2020 10:00 AM          544 contribution-props
07-12-2019 02:54 PM    <DIR>           doc
07-19-2019 09:16 AM    <DIR>           doc
07-01-2019 01:00 PM           21 dirs
01-13-2020 11:06 AM    <DIR>           doc
07-01-2019 01:00 PM           1,116 LICENSE
12-26-2019 06:18 AM           3,078 NOTICE.md
07-19-2019 09:16 AM           1,128 NuGet.Config
10-15-2019 09:57 AM    <DIR>           obj
01-13-2020 11:06 AM           111,391 OpenConsole.sln
01-13-2020 11:06 AM           1,000 packages
01-13-2020 11:07 AM    <DIR>           pkgs
07-01-2019 01:00 PM    <DIR>           README.md
01-03-2020 09:00 AM           12,978 Release
07-14-2019 05:05 PM    <DIR>           res
12-26-2019 06:18 AM    <DIR>           samples
07-01-2019 01:00 PM    <DIR>           2,766 SECURITY.md
11-01-2019 11:41 AM           83,404 standarddog.htm
01-13-2020 11:06 AM    <DIR>           src
01-13-2020 11:06 AM    <DIR>           tools
07-19-2019 09:16 AM           19 File(s)      260,046 bytes
07-14-2019 05:06 PM    <DIR>           w4
20 Dir(s)   226,359,670,016 bytes free
```

C:\Users\scilimone\GitHub\WindowsTerminal>

O poder do GitHub para a sua carreira em programação

O poder do GitHub para a sua carreira em programação



O GitHub funciona como uma rede social de pessoas programadoras, focada em compartilhamento de código.

Podemos **mostrar nosso trabalho e contribuir com o trabalho dos outros.**

A photograph of SpongeBob SquarePants standing on a beach.	1000 Instagram followers
A photograph of SpongeBob SquarePants looking sad and holding his head.	100 twitter followers
A photograph of SpongeBob SquarePants flexing his muscles.	5 reddit followers
A photograph of Squidward Tentacles looking disgruntled.	1 follower on github

O poder do GitHub para a sua carreira em programação



O GitHub funciona como uma rede social de pessoas programadoras, focada em compartilhamento de código.

Podemos **mostrar nosso trabalho e contribuir com o trabalho dos outros.**



Praticamente todo mundo que trabalha com programação tem uma conta no GitHub.

Ele é o maior host de código do mundo, com ~40 milhões de usuários e mais de 190 milhões de repositórios.

O poder do GitHub para a sua carreira em programação

O GitHub é composto basicamente por



Perfis

É o seu perfil, como um perfil no LinkedIn, perfil no Instagram, etc. Um perfil também pode ser de uma organização/empresa, etc.



Repositórios

São literalmente pastas que podem ser compartilhadas com os outros. Os repositórios possuem arquivos de código e subpastas. No GitHub, você pode criar 1 ou centenas de repositórios.

O poder do GitHub para a sua carreira em programação



O GitHub também é uma excelente **ferramenta de gestão**, em que podemos ter manuais do código, controle de versões, automações, entre outros elementos que facilitam o trabalho em equipe.

O poder do GitHub para a sua carreira em programação

git

Para usar o GitHub, precisamos do Git! Seja qual for o lugar que você irá trabalhar, Git certamente estará presente para gerir o código.



O GitHub também é uma excelente **ferramenta de gestão**, em que podemos ter manuais do código, controle de versões, automações, entre outros elementos que facilitam o trabalho em equipe.

O poder do GitHub para a sua carreira em programação

Vamos ver um perfil do GitHub =)



Cecilia Souza

Formada em jornalismo em 2011, seguiu carreira na área de formação até que em 2019 decidiu mudar de carreira. Queria se tornar programadora. Fez vários cursos, programou quase que todos os dias e agora ela é Engenheira Frontend em Berlim/Alemanha.

Agora vamos olhar o GitHub dela:
<https://github.com/cecisousa>

Obs: depois leiam o depoimento dela no nosso eBook! :D

Retomando... por que GitHub?



Para compartilhar repositórios (pastas) de código.



Um repositório (pasta) não contém somente o código, mas um histórico de tudo que ele já fez naquele código.



Ele dá visibilidade aos seus projetos, ao que você cria de código e deixa disponível lá. **Recrutadores do mundo inteiro estão sempre olhando o GitHub!**



Ele permite não só a visibilidade, mas também o trabalho em equipe, controlando as alterações no código e versões do projeto.

Iniciando um repositório

(hora de ver a prática...)

Informes

Tarefa do Dia 1 já está disponível!

Prazo final de envio da tarefa do Dia 1: quarta-feira (22/06)

Gravação da aula de hoje tentaremos disponibilizar **ainda hoje.**

Introdução ao terminal

Comando `pwd`

- PWD significa **present working directory**. Ou seja, “mostre o diretório corrente”.
- **Como utilizar:** Seu uso no terminal é muito simples, é só digitar `pwd` e pressionar enter.
- **Quando utilizar:** use ele a todo momento! Esse comando te ajuda a saber em qual pasta você está no terminal. Isso é muito importante, pois muitas vezes vocês vai precisar executar um comando para um arquivo que está dentro de uma pasta. Se você tentar executar o comando, e não estiver dentro da pasta onde está o arquivo, o comando não irá funcionar. Por isso, **utilize o comando `pwd` o tempo inteiro!**

Introdução ao terminal

Comando **ls**

- LS vem de **listar**. Ou seja, “mostre a lista do que está na pasta atual”. Quando você executa esse comando, ele mostra o que tem dentro da pasta atual que você está no terminal.
- **Como utilizar:** também é mega simples, é só digitar **ls** e pressionar enter.
- **Quando utilizar:** use ele também a todo momento! Esse comando te ajuda a saber o que tem dentro de uma pasta. Isso é muito importante, pois muitas vezes vocês vai precisar executar um comando para um arquivo que está dentro de uma pasta. Por isso, use o **ls** pra se certificar que o arquivo está realmente na pasta atual, caso contrário o comando que você quer executar não irá funcionar. Por isso, **também utilize o comando ls o tempo inteiro!**

Introdução ao terminal

Comando `cd`

- CD vem de **change directory**. Ou seja, “modifique o diretório que estou no terminal”. Quando você executa esse comando, você sai de um diretório e vai para um outro.
- **Como utilizar:**
 - `cd`: use apenas ‘`cd`’ pra você entrar na pasta raiz do seu computador. É a pasta “deus”, tudo no seu computador está dentro dela. No Windows costuma ser a pasta C/
 - `cd ..`: use `cd ..` (dois pontos) para voltar para a pasta anterior
 - `cd nomeDaSubpasta`: use para entrar numa pasta que está dentro da pasta atual
 - `cd ../../..`: volta três pastas. Você pode fazer ‘`..`/’ quantas vezes quiser. O número de vezes determinará quantos diretórios você vai voltar pra trás
 - `cd ./nomedapasta`: volta uma pasta e entra na pasta de nome ‘nomedapasta’. Pegou a lógica? Sim, você pode fazer por exemplo ‘`../../../../nomedapasta`’, ou seja, você voltou 4 pastas e entrou na pasta que está dentro dessa 4^a pasta que você voltou
 - `cd + pressionar tab 2 vezes`: **este é um truque.** Muitas vezes quando você está usando o comando `cd`, você quer saber o nome das pastas pra justamente entrar nelas digitando o nome certo. Neste caso, quando você digita ‘`cd`’ no terminal e pressiona tab 2 vezes, você executa o comando ‘`ls!`’! Ou seja, assim você consegue ver os nomes das pastas para entrar nelas corretamente. Então se você fizer `cd .. + pressionar tab 2 vezes`, você vai ver o que tem na pasta anterior e assim entrar na pasta que você quer entrar que está dentro da pasta anterior. Legal né?

Introdução ao terminal

Comando `cd`

- **Quando utilizar:** use ele para navegar nas pastas no terminal! Esse comando é muuuuito utilizado. Use-o sempre junto com o 'ls' e o 'pwd'. Esses 3 comandos é a base pra você conseguir navegar pelas pastas dentro do terminal.

Introdução ao terminal

Comando `mkdir`

- MKDIR significa **make directory**. Ou seja, “crie um diretório na pasta atual”.
- **Como utilizar:**
 - `mkdir nomedapasta`: crie o diretório ‘nomedapasta’ no diretório atual.
- **Quando utilizar:** esse é óbvio, para criar pastas. Só lembre-se sempre que quando você utilizar, você estará criando a pasta no diretório atual que você se encontra no terminal. Por isso, certifique-se com o comando ‘pwd’ que você realmente está no diretório que você quer criar a pasta.

Introdução ao terminal

Comando `touch`

- TOUCH é o verbo em inglês ‘tocar’. O significado não faz muito sentido, mas dizem que é “putting fresh fingerprints on it” (ou seja, coloque impressões digitais frescas nele). Enfim, use este comando para criar arquivos dentro do diretório atual.
- **Como utilizar:**
 - `touch nomedoarquivo`: crie um novo arquivo no diretório atual.
- **Quando utilizar:** esse é óbvio, para criar arquivos. Só lembre-se sempre que quando você utilizar, você estará criando o arquivo no diretório atual que você se encontra no terminal. Por isso, certifique-se com o comando ‘pwd’ que você realmente está no diretório que você quer criar o arquivo.

Introdução ao terminal

Comando `cat`

- CAT não tem muito significado, mas vem de concatenar. O comando é para mostrar tudo que tem dentro de um arquivo.
- **Como utilizar:**
 - `cat nome do arquivo`: mostre tudo que tem dentro de um arquivo localizado no meu diretório atual.
- **Quando utilizar:** para ver tudo que tem dentro de um arquivo.

Introdução ao terminal

Comando `head`

- HEAD é cabeçalho, ou cabeça. É um comando para mostrar o que tem de conteúdo no início de um arquivo, ou seja, na “cabeça” daquele arquivo.
- **Como utilizar:**
 - `head nome do arquivo`: mostre o início do que tem dentro de um arquivo localizado no meu diretório atual.
- **Quando utilizar:** para dar uma olhadinha no que tem dentro de um arquivo. O comando ‘cat’ mostra todo o arquivo, então se o arquivo for gigante, vai poluir seu terminal com milhões de informações. Então é isso, dar uma olhadinha no que tem dentro de um arquivo.

Introdução ao terminal

Comando grep

- O significado de grep é um pouco avançado, pois é um conhecimento que você vai poder adquirir mais lá pra frente. Mas vou falar, é **g**lobal **r**egular **e**xpression **p**rint. Significar exibir por expressão regular global. Mas esqueça esse significado por agora. O comando *grep* serve para pesquisar conteúdo dentro de um arquivo.
- **Como utilizar:**
 - **grep 'ituring' nomedoarquivo**: mostre todas as linhas do arquivo presente no diretório atual que tem a palavra *ituring*.
- **Quando utilizar:** para pesquisar conteúdo dentro de arquivos. Hoje você pode não ver muita utilidade, mas no futuro você pode fazer coisas incríveis com grep!