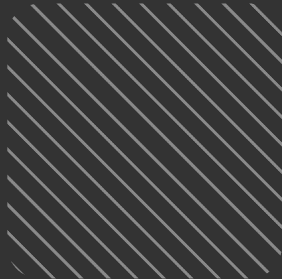




Terminal GIT & GitHub

IT BOARDING

BOOTCAMP



// Terminal - Console - CMD



O que é isto?

IT BOARDING

BOOTCAMP



O terminal é um programa que está presente em todos os Sistemas Operacionais (SO) e por meio do qual é possível dar ordens ao sistema através de linhas de comando.

IT BOARDING

BOOTCAMP

// Por que utilizar o Terminal???



- Para ter maior controle sobre o sistema operacional.
- Porque é muito comum em ambientes de desenvolvimento.
- Porque algumas linguagens de programação "exigem" isso.

Se soubermos como usar o terminal e nós nos acostumarmos com o uso em si, podemos otimizar muito nosso trabalho de programação.

// Em todos os SO temos um terminal



```
howtogeek@ubuntu: ~/Downloads
howtogeek@ubuntu:~/Downloads$ ls
file
howtogeek@ubuntu:~/Downloads$ mv file newfile
howtogeek@ubuntu:~/Downloads$ ls
newfile
howtogeek@ubuntu:~/Downloads$
```

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>powercfg /energy
Enabling tracing for 60 seconds...
Observing system behavior...
Analyzing trace data...
Analysis complete.

Energy efficiency problems were found.

5 Errors
5 Warnings
25 Informational

See C:\Windows\system32\energy-report.html for more details.
C:\Windows\system32>
```

```
Kenny — bash — 78x20
Last login: Mon Apr 13 11:46:14 on ttys000
Kennys-MacBook-Pro:~ Kenny$ mv ~/Documents/Test/TestFile-copy.rtf ~/Documents/
Test2/TestFile-copy.rtf
```

Comandos **básicos** que **devemos** saber

IT BOARDING

BOOTCAMP



// Comandos importantes



ls (no Mac e Linux mostra os arquivos da pasta em que estamos localizados. Se utilizamos o PowerShell do Windows o resultado é o mesmo)

dir (no Windows mostra os arquivos da pasta em que estamos localizados)

cd .. (nos permite retroceder a uma pasta anterior (um nível acima na estrutura de pastas))

cd nome-pasta (nos permite acessar a pasta especificada)



// Comandos importantes (diretórios)

mkdir “exemplo”

(cria uma pasta com o nome “exemplo”)

rmdir “exemplo”

(remove uma pasta com o nome “exemplo”)



// Comandos importantes (Arquivos)



touch arquivo.txt

(cria um arquivo de texto "arquivo.txt")

rm arquivo.txt

(remove um arquivo com o nome "arquivo.txt")

mv nome.txt outro.txt

(altera o nome do arquivo "nome.txt" para "outro.txt")



// Comandos importantes (Arquivos)



echo “meus dados” > arquivo.txt

(Cria arquivo.txt e insere o texto “meus dados”)

cat arquivo.txt

(mostra o texto do arquivo.txt na tela)



// Comandos importantes



clear

(limpa todo o conteúdo que havíamos escrito em nosso console / Mac e Linux ou no **PowerShell** do Windows)

cls

(limpa todo o conteúdo que havíamos escrito em nosso console / Windows)



O que é **GIT**?



git

IT BOARDING

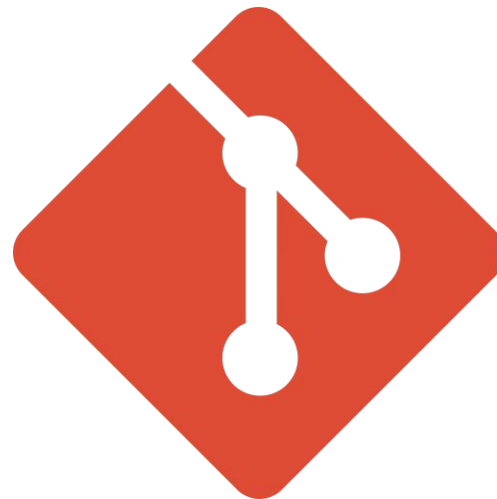
BOOTCAMP



//GIT



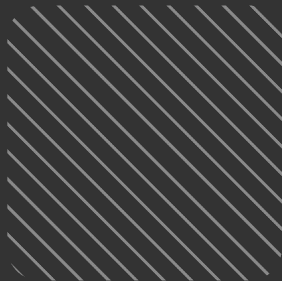
"É um **software de controle de versão** que registra as alterações feitas em um arquivo ou conjunto de arquivos ao longo do tempo"



Qual é a relação do Terminal com o **GIT**?

IT BOARDING

BOOTCAMP



//Terminal e GIT



- O terminal nos permite executar diferentes comandos GIT para gerenciar as versões dos diferentes arquivos com os quais estamos trabalhando.
- Podemos, para isso, utilizar o terminal próprio do Git ou o terminal de comandos do sistema operacional que instalamos.





git

REPOSITÓRIOS LOCAIS E REMOTOS



GitHub

IT BOARDING

BOOTCAMP



//O que são os REPOSITÓRIOS?



- Um repositório é um local onde podemos guardar e administrar diferentes arquivos. Eles nos servem como containers de informação.
- Um repositório pode ser **local** ou **remoto**:

Local

- Se executam de forma “local” em cada computador
- Não se conectam com outros repositórios nem são compartilhados.

Remoto

- Estão alojados em algum servidor externo.
- Pode ser acessado de qualquer computador ou lugar.
- Existem serviços de repositórios remotos, por exemplo: GitHub, GitLab, entre outros.



git

Repositório Local



GitHub

Repositório Remoto

IT BOARDING

BOOTCAMP



Como criamos um **REPOSITÓRIO?**



IT BOARDING

BOOTCAMP



- A primeira coisa será nos localizarmos com o terminal na pasta onde queremos criar o repositório;
- Depois escreveremos o seguinte comando:
`git init`



git init

Cria um repositório local (em nosso computador) e nos permite começar a utilizar todas as funcionalidades do GIT.

Geralmente cria uma pasta oculta a qual contém todo o repositório e suas ramificações distintas.





Obrigado.

IT BOARDING

BOOTCAMP

