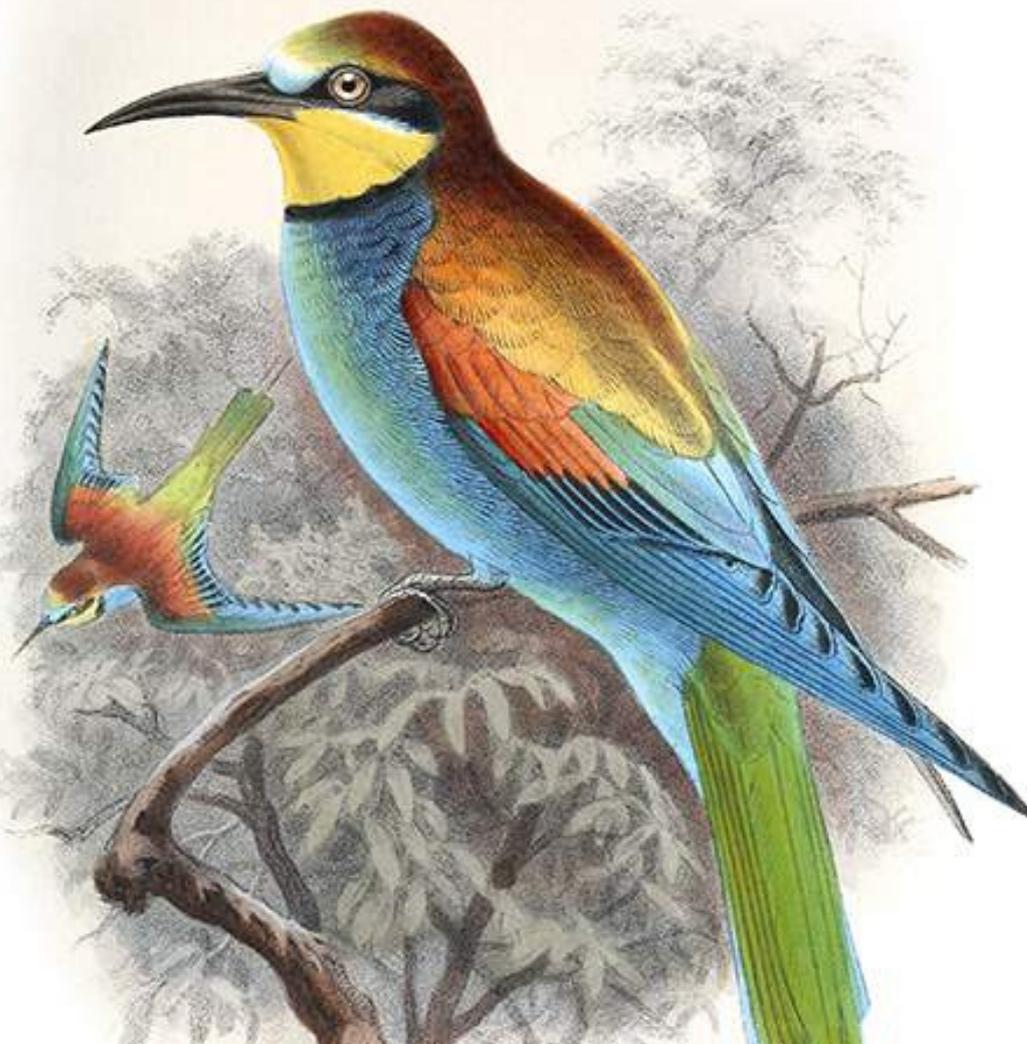


BRUNNA CROCHES

PARTE 4

# TERMINAL



Guia do  
**TERMINAL**

# ABOUT ME



*Brunna Croches*

***Developer Full Stack***

Brunna Croches é Dev FullStack, advogada e empreendedora.

Apaixonada por tech, vem adquirindo vasto conhecimento na área.

Desenvolveu projetos ricos em diversidade, buscando captar as próximas tendências e necessidades do mercado.

Neste e-book você aprenderá ou recapitulará de forma simplificada e otimizados conceitos de programação feito por ela.

*let's share*

# SUMMARY



---

**MKDIR**

**2.3**

---

**COMO EXIBIR  
PASTAS E  
ARQUIVOS  
OCULTOS NO PC,  
MAC OU PENDRIVE?**

**2.4**

---

**GIT LINHA DE  
COMANDOS**

**3.0**

---

**GITHUB NO  
TERMINAL**

**3.1**

---

## 2.3 MKDIR

| Aulas<br>Dani   |  |
|---|--|
| <code>mkdir</code>  | criar pasta  |
| <code>echo</code>   | repetir o que você escreveu<br>expressar o que vai ser dito<br>em seguida<br><br>não e um coando é um<br>conteúdo          |
| <code>cat</code>  | le o arquivo de texto  |
| <code>echo 'frango é bom' &gt; foo.txt</code>   | colocar um 'texto' dentro de<br>um arquivo   |
| como quebrar linha no<br>terminal<br><br><code>echo "linha1"</code>                       | <code>echo "adicionar" &gt;&gt; foo.txt</code>   |
| <code>echo &gt; (uma setinha)</code>  | substitui tudo   |
| <code>echo "linha2" &gt;&gt;</code>   | adiciona linha para baixo  |
| <code>echo -A(fter)</code><br><code>echo -B(efore)</code><br><code>echo -C(ontext)</code> | mostra linha : depois<br>mostra linha: antes<br>mostra linha : antes e depois  |
|   | " "<br>` `<br>' '  |
| <code>.</code>  | dizer que estou me referindo<br>ao diretório que estou no<br>momento   |
| <code>ps</code>   | como saber se os processos<br>do terminal está rodando   |
| <code>rm</code>   | comando que apaga  |
| <code>GREP</code>   | <code>grep lorem foo.tx</code> (letra<br>minúscula ele não acha )<br>se eu não sei :<br><code>grep -i lorem foo.txt</code> |
| <code>grep qualquer foo.txt</code>  | <code>grep</code> : procurar<br>palavra/arquivo dentro da<br>pasta/arquivo txt   |
| <code>-i</code>   | pesquisa independe letra<br>maiúscula ou minúscula   |
| <code>-E</code>   |  |
| <code>-r</code>   | entra em toda estrutura de<br>diretórios e faz isso em todos<br>os arquivos  |

|  |  |
|--|--|
| (recursivamente)<br>procurar em toda a árvore                      | "pinta a casa"   |
| grep -r today .  | procurar todas palavras<br>"today" ( grep + - r ) a partir<br>do diretório onde eu me<br>encontro ( __ . )   |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| mostrar em qual diretório ele<br>está                              | cd   |
| demonstra quais são as<br>pastas que existem no meu<br>mac         | ls   |
| sair do terminal sem fechar o<br>terminal                          | control + z  |
| mostra os diretorios e cá<br>arquivo na pasta                      | ls - l   |
| limpar   | clear  |
| entrar em qualquer diretório<br>que vocês queiram dentro do<br>mac | <b>ls &gt;</b><br><b>cd movies &gt;</b><br>(saber se bem pasta)<br><b>&gt; ls</b><br>criar uma pasta dentro dessa<br>pasta<br><b>&gt; mkdir +</b><br>(nome da pasta que vou<br>querer criar) " <b>testa</b> "<br>> pra saber se a pasta foi<br>criada mesmo<br><b>&gt; ls - l</b><br><b>pasta teste criada</b> |
| acessar pasta que acabamos<br>de criar                             | cd teste/<br>( / > pq ja estamos dentro da<br>pasta movies ) e agora<br>estamos dentro da pasta teste  |
| apagar essa pasta  | pwd  |
| voltar ..  | cd .. (ele vai retroceder pra<br>pasta anterior pasta movies)  |
| listar as pastas   | ls - l   |
| diretório pasta  | rm-teste/  |
| pasta apagada  | rmdir teste/   |
|  |  |

|                                      |                       |
|--------------------------------------|-----------------------|
| ver se a pasta realmente foi apagada | ls -l                 |
| cd - procurar                        |                       |
| open .                               | abrir na pasta finder |

## 2.4 - Como exibir pastas e arquivos ocultos no PC, Mac ou pen drive

### PARA EXIBIR PASTAS E ARQUIVOS OCULTOS

1. Entre em **Aplicativos > Utilitários > Terminal**;
2. Envie o comando

```
defaults write com.apple.finder AppleShowAllFiles YES; killall Finder  
aperte Enter
```

3. Pronto! Agora as pastas e arquivos ocultos estarão visíveis.

### PARA OCULTAR PASTAS E ARQUIVOS OCULTOS

1. Para esconder os arquivos novamente, digite este comando no Terminal:

```
defaults write com.apple.finder AppleShowAllFiles NO; killall Finder
```

2. e aperte **Enter**.

HTML Content

## Como exibir pastas e arquivos ocultos no PC, Mac ou pen drive

Veja como mostrar ou esconder arquivos do seu PC sem muitos esforços



Por **Jean Prado**  
3 anos atrás

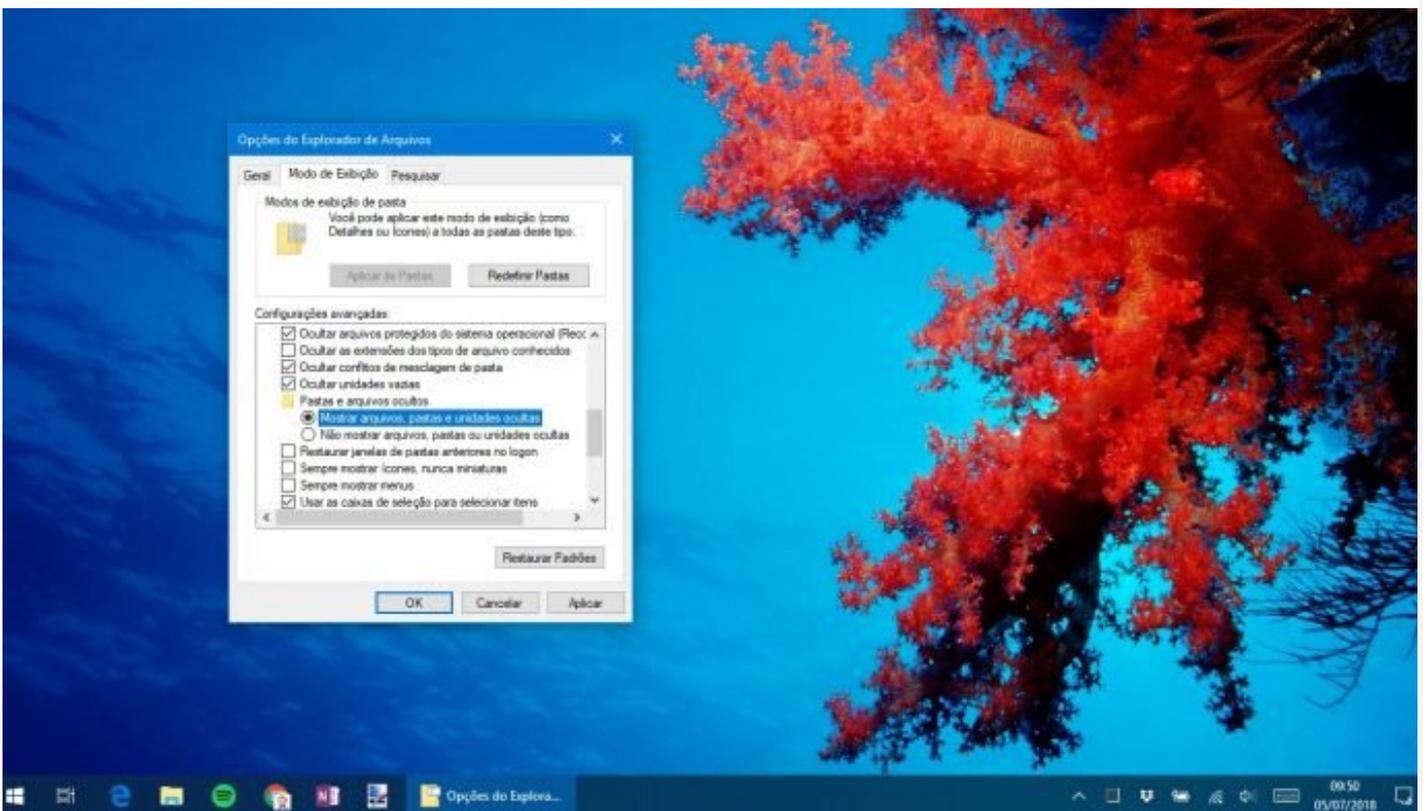
Se você quer **exibir arquivos ocultos no PC**, seja [Windows](#), Mac ou até mesmo em um pen [drive](#), consegue fazer isso facilmente alterando algumas configurações do computador. Seja para ver **pastas ocultas por conta de um vírus** ou somente para **mostrar arquivos ocultos do sistema** que não são visíveis, o processo não é complicado.

- [Como formatar um pen drive no Windows ou Mac](#)
- [Como remover vírus que oculta pastas do pen drive](#)



Vale o lembrete: arquivos do sistema estão ocultos por um motivo, e tome cuidado ao lidar com eles. Se você renomear ou excluir algum sem querer, pode afetar o funcionamento do computador. Tenha sempre um [backup](#) dos seus arquivos, principalmente de um pen drive que você usa muito. Se ele estiver com vírus, veja [como remover vírus que oculta pastas do pen drive](#).

## Como exibir arquivos ocultos no PC e pen drive



## Se você usa Windows 7...

Clique no botão iniciar e entre no **Painel de Controle**;

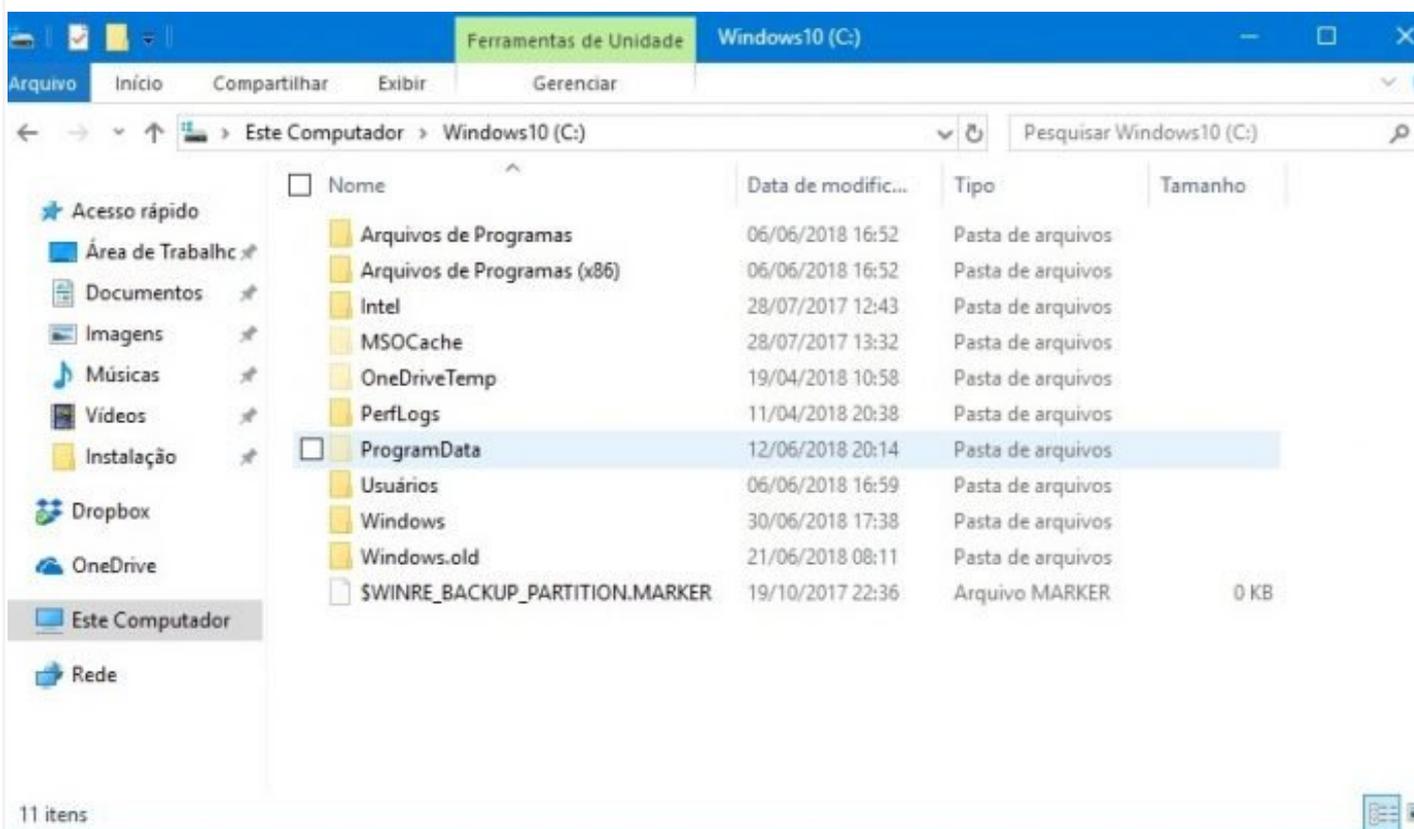
Acesse **Aparência e Personalização** e clique em **Opções de Pasta**;

Entre na aba **Exibir**;

Procure Configurações avançadas e selecione a opção **Mostrar arquivos, pastas e unidades ocultas**;

Para finalizar, clique em **OK**;

## Se você usa Windows 10...



Na caixa de pesquisa dentro da barra do Windows, procure por **pasta**;

Clique em **Mostrar pastas e arquivos ocultos**;

Em "Pastas e arquivos ocultos", marque a opção **Mostrar arquivos, pastas e unidades ocultas**;

Para finalizar, clique em **OK**;

## Como exibir arquivos ocultos no Mac

No Mac, não existe forma de exibir arquivos ocultos acessando as Preferências do Sistema. No entanto, o processo é bem mais simples do que no PC. Olha só:



Use o atalho de teclado **Command** (⌘) + **Shift** + **.** (sim, o ponto final) dentro do Finder;

Pronto! Agora o Finder deve exibir as pastas e arquivos ocultos do sistema. Para ocultá-los novamente, pressione o mesmo atalho.

Não deu certo?

```
jeanprado — -bash — 80x24
Last login: Thu Jul 5 12:22:53 on ttys000
Jeans-MBP-6:~ jeanprado$ defaults write com.apple.finder AppleShowAllFiles YES; killall Finder
Jeans-MBP-6:~ jeanprado$ defaults write com.apple.finder AppleShowAllFiles NO; killall Finder
Jeans-MBP-6:~ jeanprado$
```

Se você não conseguiu da primeira forma, pode tentar no aplicativo de Terminal.

Entre em **Aplicativos > Utilitários > Terminal**;

Envie o comando `defaults write com.apple.finder AppleShowAllFiles YES; killall Finder` e aperte **Enter**;

Pronto! Agora as pastas e arquivos ocultos estarão visíveis.

Para esconder os arquivos novamente, digite este comando no Terminal:

```
defaults write com.apple.finder AppleShowAllFiles NO; killall Finder e aperte Enter.
```

TB RESPONDE

## 3.0 GIT - LINHAS DE COMANDO

|  |
|--|
|  |
| <b>Iniciar o GIT</b>   |
| <code>git init</code>  |
| <b>Verifica se houve alguma alteração na working área</b>                          |
| <code>git status</code>  |
| <b>Adiciona o arquivo file.js ao stage</b>   |
| <code>git add file.js</code>   |
| <b>Committa as alterações no repositório local</b>                                 |
| <code>git commit -m "criação do file.js"</code>                                    |
| <b>Define o branch 'ProjetoTeste' como a área de trabalho (working area) atual</b> |
| <code>git checkout ProjetoTeste</code>   |
|  |
| <b>Inicia o Git</b>  |
| <code>git init</code>  |
| <b>Verifica se houve alguma alteração na working área</b>                          |
| <code>git status</code>  |
| <b>Adiciona o arquivo file.js ao stage</b>   |
| <code>git status</code>  |
| <b>Adiciona o arquivo file.js ao stage</b>   |
| <code>git add file.js</code>   |
| <b>Committa as alterações no repositório local</b>                                 |
| <code>git commit -m "criação do file.js"</code>                                    |
| <b>Adicionar um arquivo ao stage</b>   |
| <code>git add</code>   |
| <code>git checkout</code>  |
| <b>Conteúdo dos dois branches seja unificado</b>                                   |
| <code>git merge</code>   |
| <b>Definir um diretório como sua área de trabalho</b>                              |
| <code>git init</code>  |

## Remover um item do stage antes de commitar

```
git rm --cached file.html
```

## Remover arquivo

```
git rm --cached dados.php
```

```
git restore --staged dados.php
```

```
git reset HEAD dados.php
```

## Envia as alterações do stage para o repositório local

```
git commit -m "Versão inicial"
```

## Acessa o histórico de alterações

```
git log
```

## Adiciona um marcador ao commit especificado pelo hash

```
git tag 1.0.0 a98368ab880ca9079cf587de851b989f09c74fbb
```

## Baixar dados do repositório externo configurado previamente

```
git push
```

## Adicionar todas as alterações ao stage

```
git init
```

## Commitar para o servidor local

```
git add
```

## Enviar para o repositório externo são executados

```
git pull
```

## Para testar uma nova funcionalidade ou biblioteca no projeto sem afetar o conteúdo principal

```
git branch
```

```
git config
```

## Configura usuário

```
git config --global user.name "Fulano da silva"
```

## Configura e-mail

```
git config --global user.email e-mail@exemplo.com
```

## Criar branch 'ProjetoLoja'

```
git branch ProjetoLoja
```

## Definir o novo branch como padrão

```
git checkout ProjetoLoja
```

## Clonar

```
git clone https://github.com/meuusuario/meu\_repositorio.git
```

## Baixa todas as novidades existentes no repositório externo configurado previamente;

```
git pull
```

## Verifica se houve alterações na working area do projeto;

```
git status
```

## Adiciona todas as alterações do diretório atual ao stage;

```
git add .
```

## Publica as alterações no repositório local

```
git commit -m "nova home"
```

## Envia todas as alterações commitadas para o repositório externo.

```
git push
```

## Publicando o projeto

```
git commit -m "nova home"
```

## Criar branch chamado "Teste"

```
git branch Teste
```

## Definir como área de trabalho

```
git checkout Teste
```

## Criar sua área de trabalho (working area) / inicializar o git

```
git init
```

## Clonar o novo projeto

```
git clone https://github.com/meusuario/novo\_projeto.git
```

## Criar um branch chamado 'ProjetoNovoDev'

```
git branch ProjetoNovoDev
```

## Checar se existe alguma modificação na working area do projeto

```
git status
```

## Adicionar todas as alterações do diretório atual ao stage

```
git add .
```

## Commitar tudo para o repositório local

```
git commit -m "bug fix"
```

## Adicionar todo conteúdo alterado no diretório local ao stage.

```
git add .
```

```
git add -A
```

```
git add --all
```

1- acessar a pasta no terminal

2- git add .

3- git commit -m + nome da pasta

4- git push origin + nome da pasta

git push --set-upstream origin backend

1- acessar a pasta no terminal

2- git init

3- gh repo create

4- git add .

5- git commit -m "mensagem de commit"

6- git push --set-upstream origin master(nome da branch)

git push --set-upstream origin master

```
(base) brunna@MacBook-Pro-de-Brunna daniel % git push --set-upstream origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 220 bytes | 220.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:brunnacroches/daniel.git
 * [new branch] master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
(base) brunna@MacBook-Pro-de-Brunna daniel %
```

## ATUALIZANDO OS ARQUIVOS

atualizar arquivos

git add . = add todos

|   |
|---|
| git add + nome do arquivo = add somente a pasta chamada   |
|   |
| git ignore = é um arquivo que você cria para não subi-los / especifique quais arquivos não vão ser preparados para commit |
|   |
| dock mostrar/ocultar atalho: command + alt + d  |
|   |

|  |                                       |
|--|---------------------------------------|
|  |                                       |
|  | saber como está configurado o meu git |
| git config --list                              | mudar o username                      |
| git config --global user.name                  | mudar email                           |
| git config --global <a href="#">user.email</a> | saber login                           |
| git log  | git status                            |
|  |                                       |

### Incluindo um novo arquivo

|   |  |
|---|--|
| git add .                               |  |
| git commit -m "Incluindo novo arquivo " |  |
| git log                                 | mostra quem fez o commit aqui atualizado |

### SINCRONIZAR COM O REPOSITÓRIO REMOTO

|   |  |
|---|--|
| no gitHub >   | criar um repositório novo                              |
| create a new repository   |  |
| em code : HTTPS   | https://github.com/brun<br>nacroches/apinumber.g<br>it |
| git remote add origin<br>https://github.com/brun<br>nacroches/apinumber.g<br>it |  |
|   |  |
| gh repo create  |  |
|   |  |

GitHub CLI > tem funções para dar comandos para o git que são exclusivos do github e não do git

## 3.1 GITHUB NO TERMINAL

### SUBIR PROJETO REMOTO PARA O GITHUB PELO TERMINAL / LINHAS DE COMANDO

1- acessar a pasta no terminal

2- git init

3- **gh** repo create

4- git add .

5- git commit -m "mensagem de commit"

6- git push --set-upstream origin master(nome da branch)

git push --set-upstream origin master

```
(base) brunna@MacBook-Pro-de-Brunna daniel % git push --set-upstream origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 220 bytes | 220.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:brunnacroches/daniel.git
 * [new branch] master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
(base) brunna@MacBook-Pro-de-Brunna daniel %
```

### ATUALIZANDO OS ARQUIVOS

atualizar arquivos

git add . = add todos

git add + nome do arquivo = add somente a pasta chamada

git ignore = é um arquivo que você cria para não subi-los / especifique quais arquivos não vão ser preparados para commit

dock mostrar/ocultar atalho: command + alt + d

# CONTACT



*Brunna Croches*

*Developer Full Stack*



*brunnacroches.dev*



*linkedin.com/brunnacroches*



*github.com/brunnacroches*



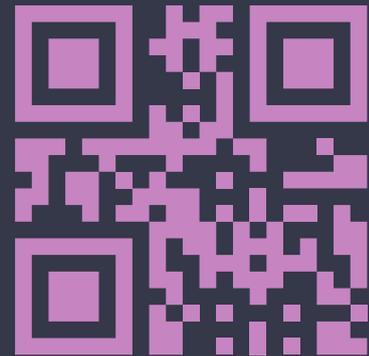
*@brunnacroches.dev*



*discord.com/brunnacroches*



*brunnacroches@gmail.com*



*let's share*