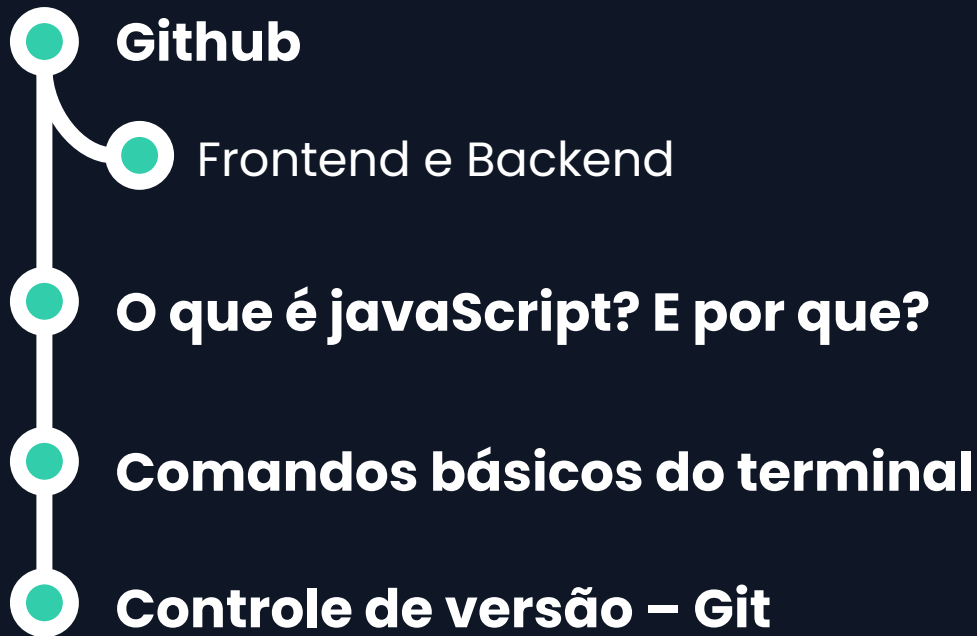




Bootcamp Full Stack Developer

Aula 1 – Construindo a base de tudo



O que é programação?

Programa ação

O que é programação?

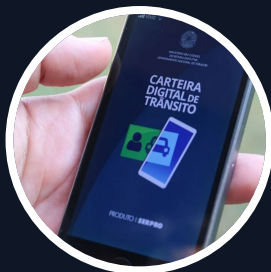


Resolver problemas utilizando instruções que podem executadas por um computador

O que é programação?



O que é programação?



O que é programação?

Os vários papéis em programação...



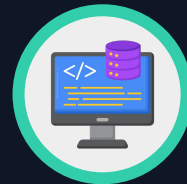
Engenharia ou ciência de dados



Engenharia de Inteligência Artificial



Desenvolvimento Frontend



Desenvolvimento Backend



Desenvolvimento Mobile

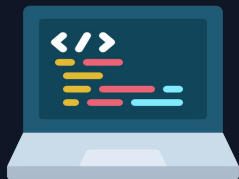


Engenharia DevOps

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

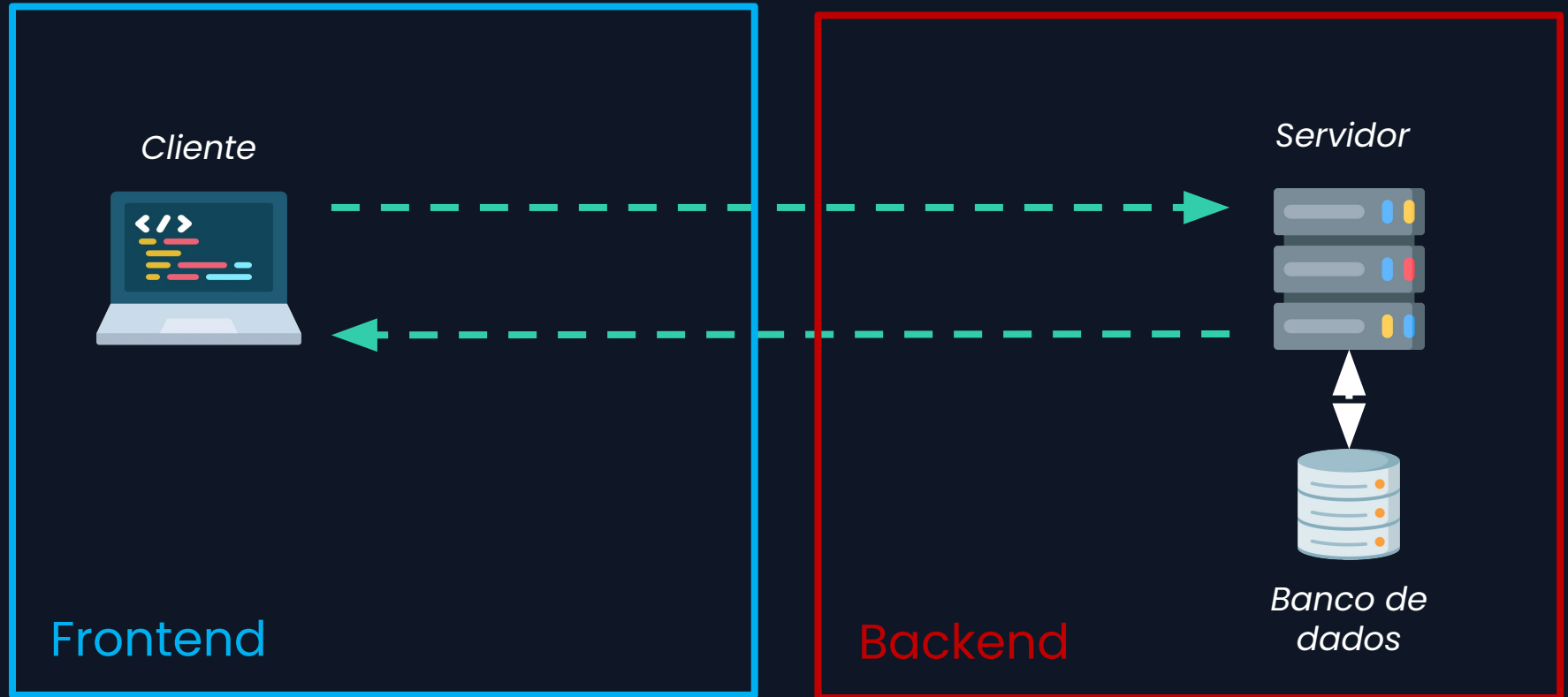
Frontend e Backend

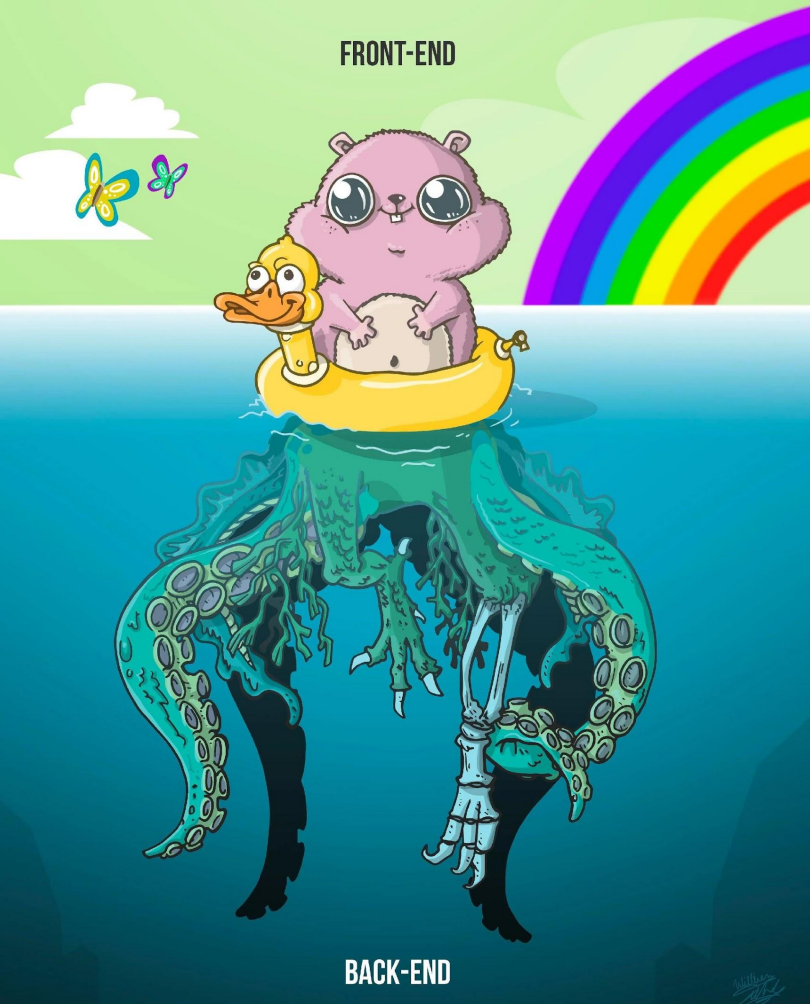
Cliente



`https://site.com.br`







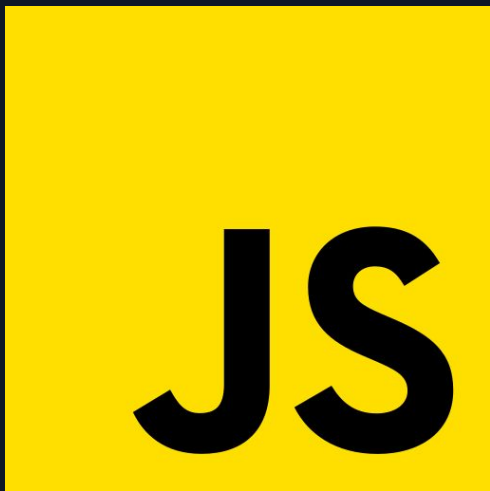
FRONT



BACK

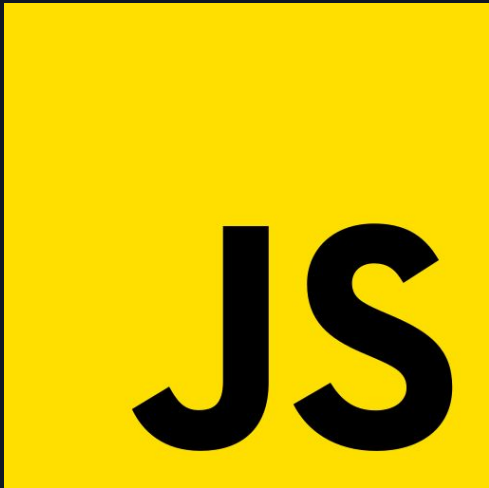


O que é Javascript ? E por que?



- Linguagem de programação (Web)
- Alto nível
- Interpretada
- Baseada em protótipo
- Dinamicamente tipada
- Multi-paradigma (imperativo, funcional e Orientado a objetos)
- Versátil (mobile, desktop, TVs, smartwatches, jogos)

E por que?

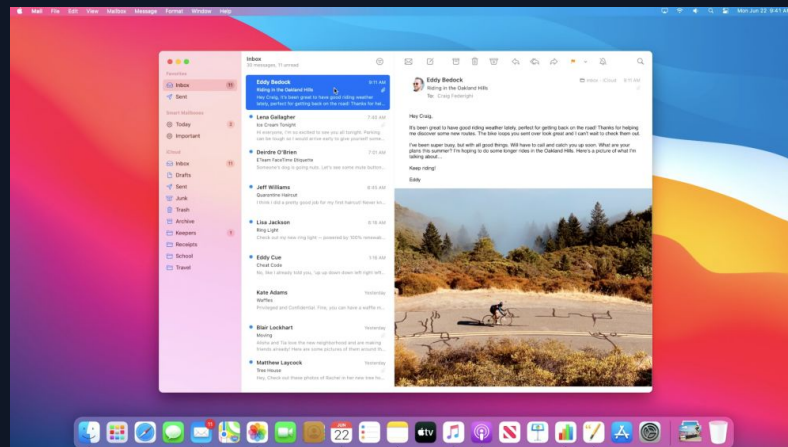
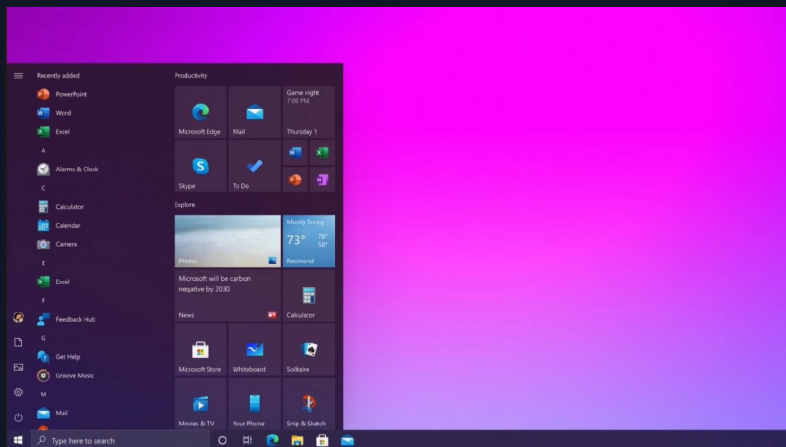


- Fácil para iniciar
- 99% dos sites da web usam JavaScript
- Consolidada (Netscape - 1995 e ECMA International - 1997)
- Em constante evolução (ES6)
- Bem documentada

<https://survey.stackoverflow.co/2022/#most-popular-technologies-language>

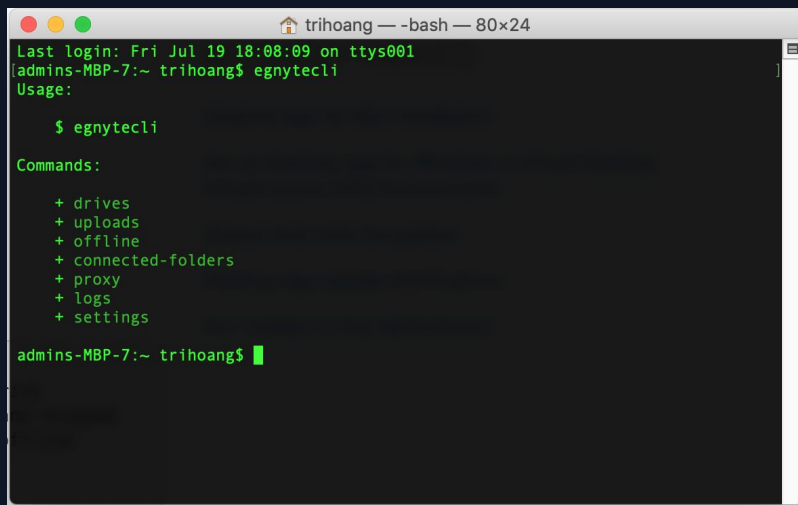
Comandos básicos do terminal

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



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



```

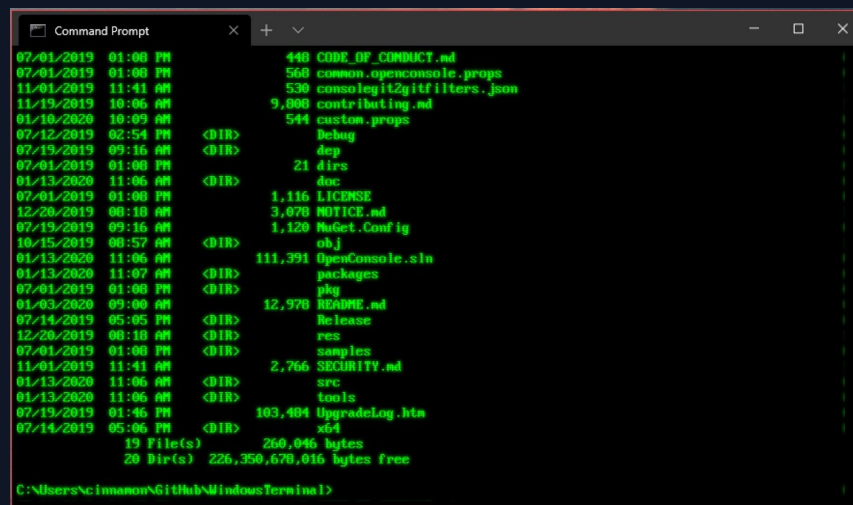
trihoang — -bash — 80x24
Last login: Fri Jul 19 18:08:09 on ttys001
[admins-MBP-7:~ trihoang$ egnytecli
Usage:

    $ egnytecli

Commands:

    + drives
    + uploads
    + offline
    + connected-folders
    + proxy
    + logs
    + settings

admins-MBP-7:~ trihoang$
  
```



```

Command Prompt
07/01/2019 01:00 PM          440 CODE_OF_CONDUCT.md
07/01/2019 01:00 PM          560 common.openconsole.props
11/01/2019 11:41 AM          530 consolegit2gitfilters.json
11/19/2019 10:06 AM          9,000 contributing.md
01/10/2020 10:09 AM          544 custom.props
07/12/2019 02:54 PM        <DIR>      Debug
07/19/2019 09:16 AM        <DIR>      dep
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
12/20/2019 00:10 AM          3,078 NOTICE.md
07/19/2019 09:16 AM          1,120 theGet-Config
10/15/2019 00:57 AM        <DIR>      obj
01/13/2020 11:06 AM        111,391 OpenConsole.sln
01/13/2020 11:07 AM        <DIR>      packages
07/01/2019 01:00 PM        <DIR>      pkg
01/03/2020 09:00 AM          12,978 README.md
07/14/2019 05:05 PM        <DIR>      Release
12/20/2019 00:10 AM        <DIR>      res
07/01/2019 01:00 PM        <DIR>      samples
11/01/2019 11:41 AM          2,766 SECURITY.md
01/13/2020 11:06 AM        <DIR>      src
01/13/2020 11:06 AM        <DIR>      tools
07/19/2019 01:46 PM        103,404 UpgradeLog.htm
07/14/2019 05:06 PM        <DIR>      x64
          19 File(s)          260,046 bytes
          20 Dir(s)  226,350,670 bytes free

C:\Users\cinamon\Git\Hub\WindowsTerminal>
  
```

O propósito do terminal é este.

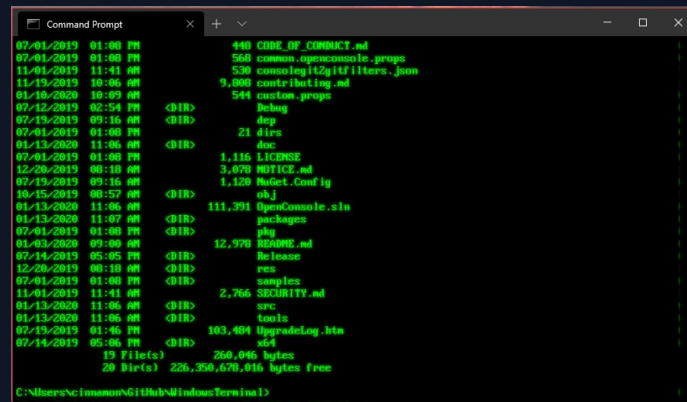
No Windows, o terminal padrão não é do tipo Unix/Linux. Por conta disso, vamos utilizar o Git Bash como terminal no Windows. O Git Bash é no padrão Unix/Linux.



Por que usar um terminal do tipo Unix/Linux? Você vai entender melhor lá no Mês 3, mas a razão é porque [mais de 96% dos principais servidores do mundo são do tipo Linux](#). Então não faz sentido aprender um terminal que quase ninguém utiliza.

Introdução ao terminal

Interagimos com o terminal por meio dos
comandos de terminal



```

Command Prompt
07/01/2019 01:00 PM 440 CODE_OF_CONDUCT.md
07/01/2019 01:00 PM 560 common.openconsole.props
11/01/2019 11:41 AM 530 consolelogit2gitfilters.json
11/13/2019 10:06 AM 9,000 contributing.md
01/10/2020 10:09 AM 544 custom.props
07/12/2019 02:54 PM <DIR> debug
07/13/2019 09:16 AM <DIR> dep
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
12/20/2019 00:10 AM 3,070 NOTICE.md
07/13/2019 09:16 AM 1,120 NuGet.Config
10/15/2019 00:57 AM <DIR> obj
01/13/2020 11:06 AM 111,391 OpenConsole.sln
01/13/2020 11:07 AM <DIR> packages
07/01/2019 01:00 PM <DIR> pkg
01/03/2020 09:00 AM 12,970 README.md
07/14/2019 05:05 PM <DIR> Release
12/20/2019 00:10 AM <DIR> res
07/01/2019 01:00 PM <DIR> samples
11/01/2019 11:41 AM 2,766 SECURITY.md
01/13/2020 11:06 AM <DIR> src
01/13/2020 11:06 AM <DIR> tools
07/13/2019 01:46 PM 103,404 Upgradelog.htm
07/14/2019 05:06 PM <DIR> x64

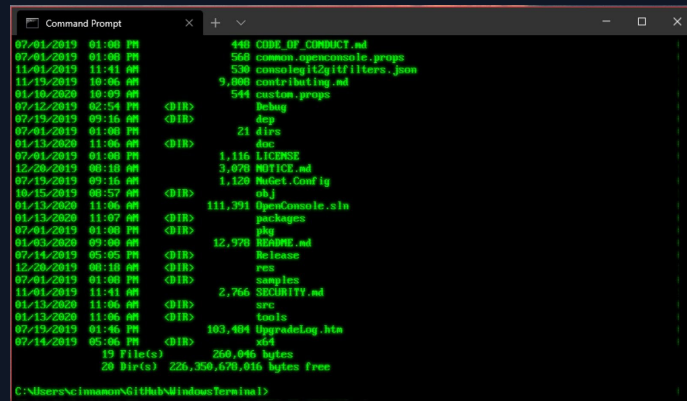
19 File(s) 260,046 bytes
28 Dir(s) 226,350,670,016 bytes free

C:\Users\cimmason\Git\Hub\WindowsTerminal>

```

Hora da prática!

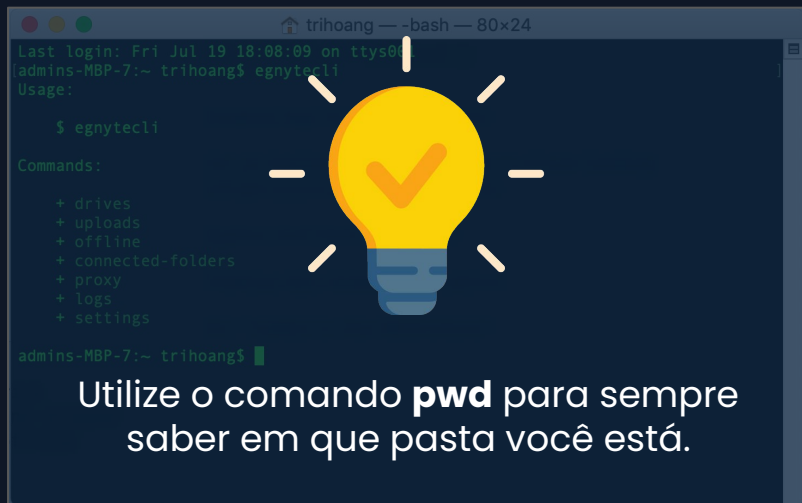
- **Principais comandos nativos do terminal:**
 - `pwd` – mostrar em que pasta estamos
 - `ls` – listar tudo que está dentro de uma pasta
 - `cd` – mudar de diretório
 - `mkdir` – criar uma pasta/dirertório
 - `touch` – criar um arquivo
 - `cat` – mostrar todo o conteúdo do arquivo
 - `head` – mostrar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



```
Command Prompt
07/01/2019 01:00 PM 440 CODE_OF_CONDUCT.md
07/01/2019 01:00 PM 560 common.openconsole.progs
11/01/2019 11:41 AM 530 consolegit2gitfilters.json
11/13/2019 10:06 AM 9,000 contributing.md
01/10/2020 10:09 AM 544 custom.progs
07/12/2019 02:54 PM <DIR> debug
07/13/2019 09:16 AM <DIR> dep
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
12/20/2019 00:10 AM 3,070 NOTICE.md
07/13/2019 09:16 AM 1,120 NuGet.Config
10/15/2019 00:57 AM <DIR> obj
01/13/2020 11:06 AM 111,391 OpenConsole.sln
01/13/2020 11:07 AM <DIR> packages
07/01/2019 01:00 PM <DIR> pkg
01/03/2020 09:00 AM 12,970 README.md
07/14/2019 05:45 PM <DIR> Release
12/20/2019 00:10 AM <DIR> res
07/01/2019 01:00 PM <DIR> samples
11/01/2019 11:41 AM 2,766 SECURITY.md
01/13/2020 11:06 AM <DIR> src
01/13/2020 11:06 AM <DIR> tools
07/13/2019 01:46 PM 103,404 UpgradedLog.htm
07/14/2019 05:06 PM <DIR> x64

19 File(s) 260,046 bytes
20 Dir(s) 226,350,670,016 bytes free

C:\Users\c:\Users\N6\Documents\Terminal\
```



```

trihoang — -bash — 80x24
Last login: Fri Jul 19 18:08:09 on ttys0
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.

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

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.

Controle de versão – conhecendo o Git



Trabalho Ciências



Trabalho.doc



Trabalho_v1.doc



Trabalho_v2.doc



Trabalho_v3_revisado.doc



Trabalho_vfinal.doc



Trabalho_vfinal_formatado.doc

O que mudou de um arquivo pra outro ?



Trabalho.doc



Trabalho_v1.doc



Trabalho_v2.doc



Trabalho_v3_revisado.doc



Trabalho_vfinal.doc



Trabalho_vfinal_formatado.doc

Como desfazer somente as modificações da versão 1?



Trabalho.doc



Trabalho_v1.doc



Trabalho_v2.doc



Trabalho_v3_revisado.doc



Trabalho_vfinal.doc



Trabalho_vfinal_formatado.doc

**Ih, é um trabalho
em grupo, quem
modificou o que ?**



Trabalho.doc



Trabalho_v1.doc



Trabalho_v2.doc



Trabalho_v3_revisado.doc



Trabalho_vfinal.doc



Trabalho_vfinal_formatado.doc

Uma aplicação web costuma ser organizada da seguinte forma ...



Projeto-cliente-alfa



assets



logo.png



banner.jpg



scripts



main.js



utils.js



index.html



styles



header.css



footer.css



main.css



reset.css

**O que
mudou?**

**Como voltar
para uma
determinada
versão?**

**Quem
modificou o
que ?**

Controlador de versão





Mão no Código!

✓ Comandos git

git



✓ Git instalado

```
git version  
ou  
git --version
```



✓ Ajuda do git

```
git help <commando>
```

ou

```
git <commando> --help
```



Como ativar o git no meu projeto ?



Projeto

Como ativar o git no meu projeto ?



Projeto



Inicializar o git

Como ativar o git no meu projeto ?



Projeto



Inicializar o git

```
git init
```



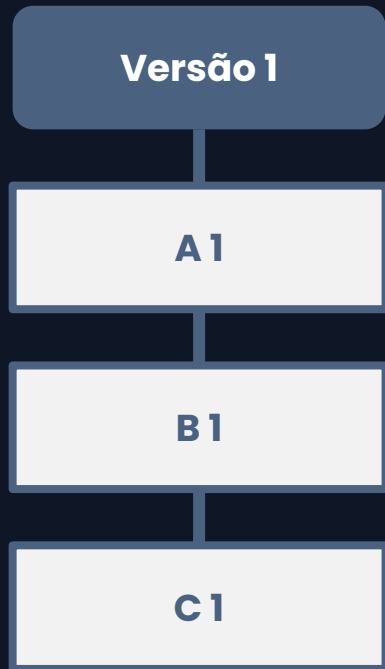
Como ativar o git no meu projeto ?



Como ativar o git no meu projeto ?



Como funciona ?



Como funciona ?



Como funciona ?



Como funciona ?





Configurar o git

Global ou local?



Configurar o git

Global ou local?

Global!



Configurar o git

Global ou local?

Global!

```
git config --global user.name "Seu Nome"
```





Configurar o git

Global ou local?

Global!

```
git config --global user.name "Seu Nome"
```

```
git config --global user.email "seu@email.com"
```



✓ Conferir configuração

```
git config user.name
```

```
git config user.email
```



✓ Conferir configuração

```
git config user.name
```

```
git config user.email
```

```
git config --list
```



Fluxo de trabalho Git

Fluxo de trabalho Git



*Criar repositório
local
e inicializar(1x)*

git init

Máquina local
(git no seu computador)

Fluxo de trabalho Git



*Criar repositório
local
e inicializar(1x)*

`git init`

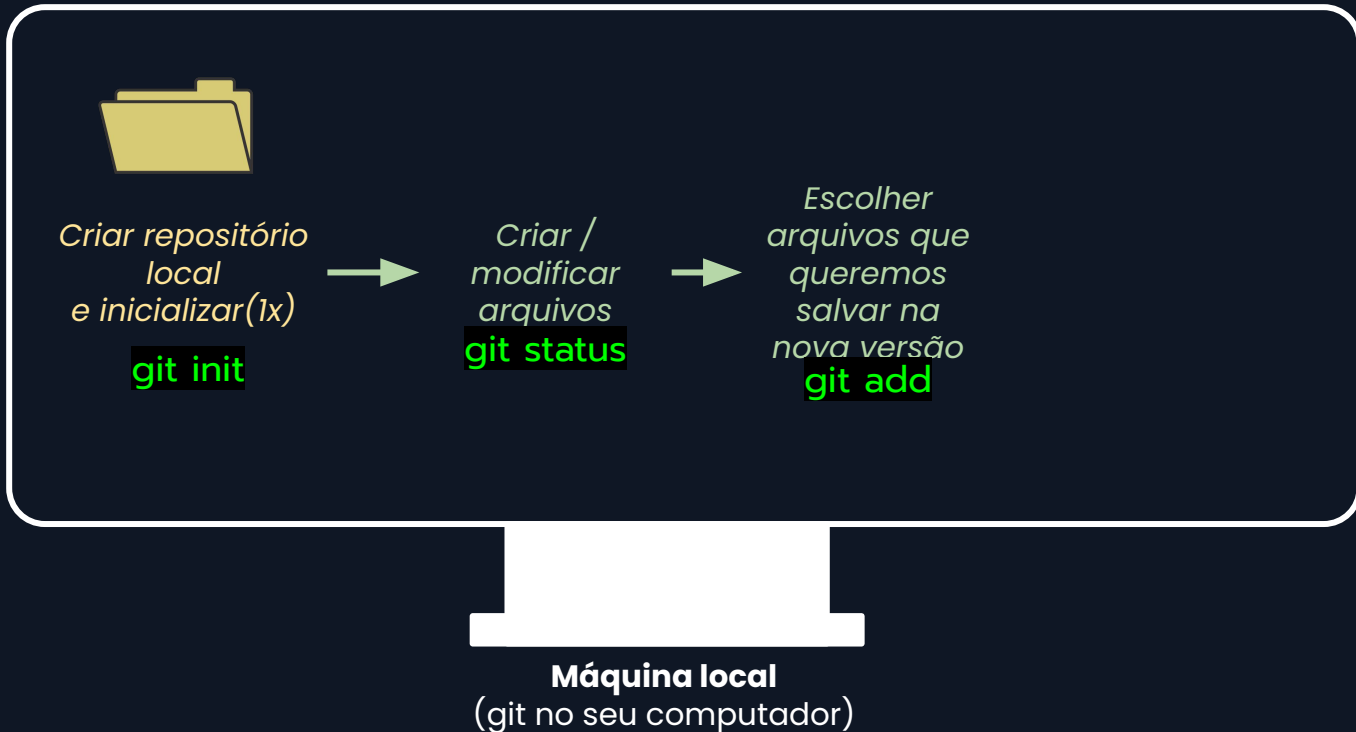


*Criar /
modificar
arquivos*

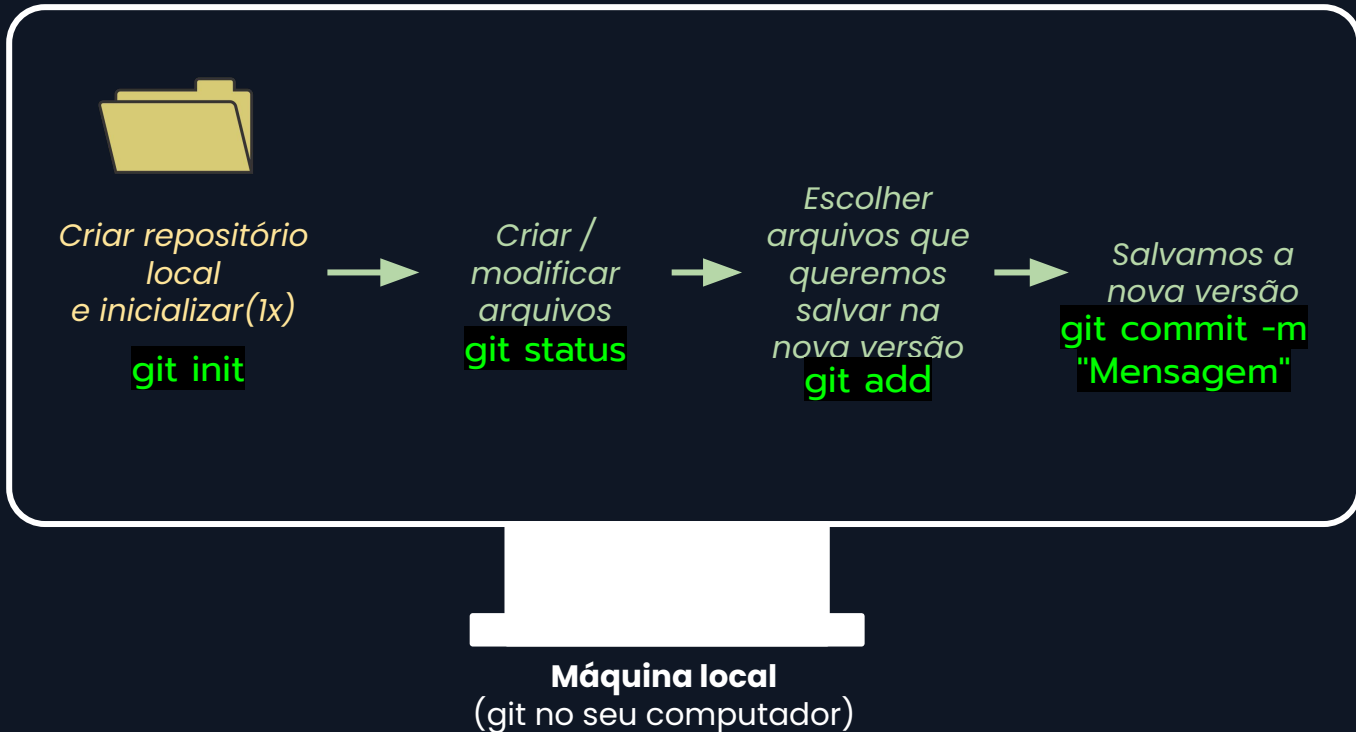
`git status`

Máquina local
(git no seu computador)

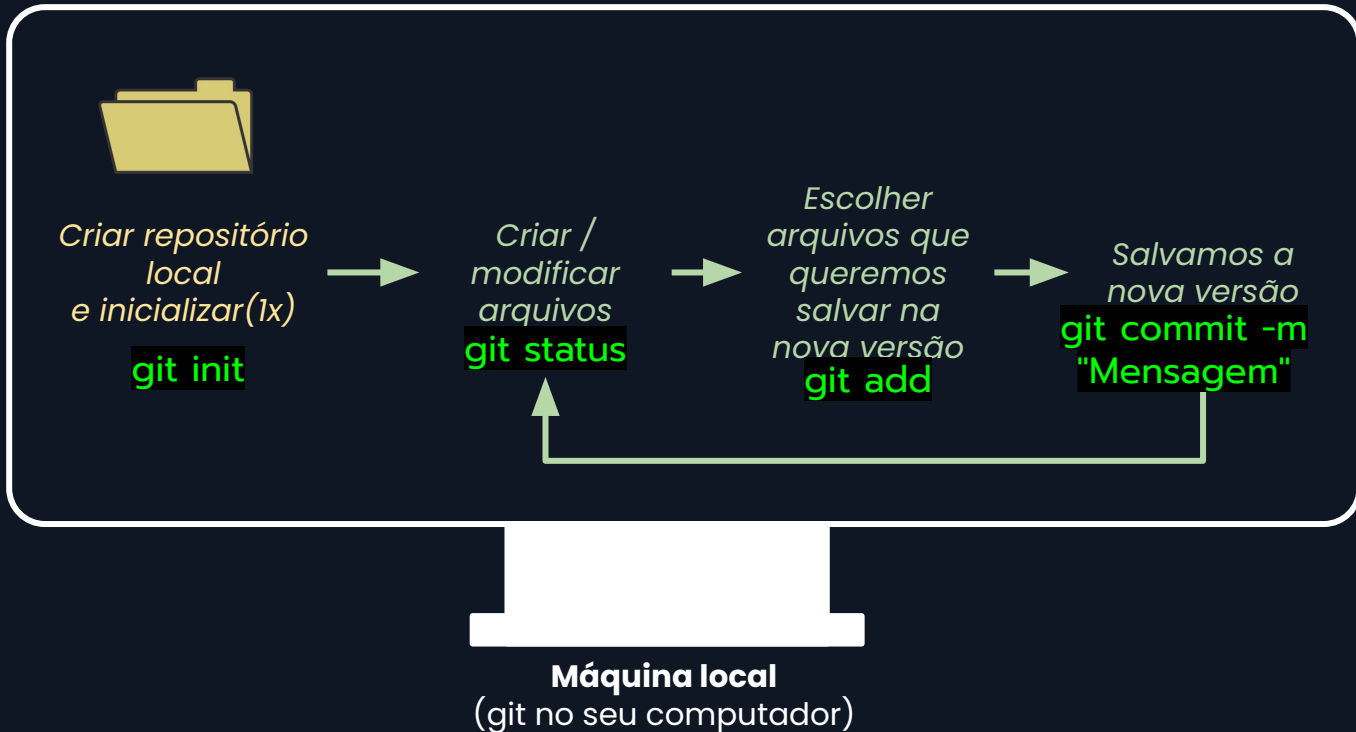
Fluxo de trabalho Git



Fluxo de trabalho Git



Fluxo de trabalho Git



✓ Vendo o histórico de modificações

```
git log
```

```
git log --oneline
```



✓ Vendo a diferença entre os arquivos

```
git diff <hashVersaoA> <hashVersaoB>
```



✓ Desfazendo coisas

```
git restore <arquivo>  
git restore --staged <arquivo>  
git reset (versões antigas)
```

