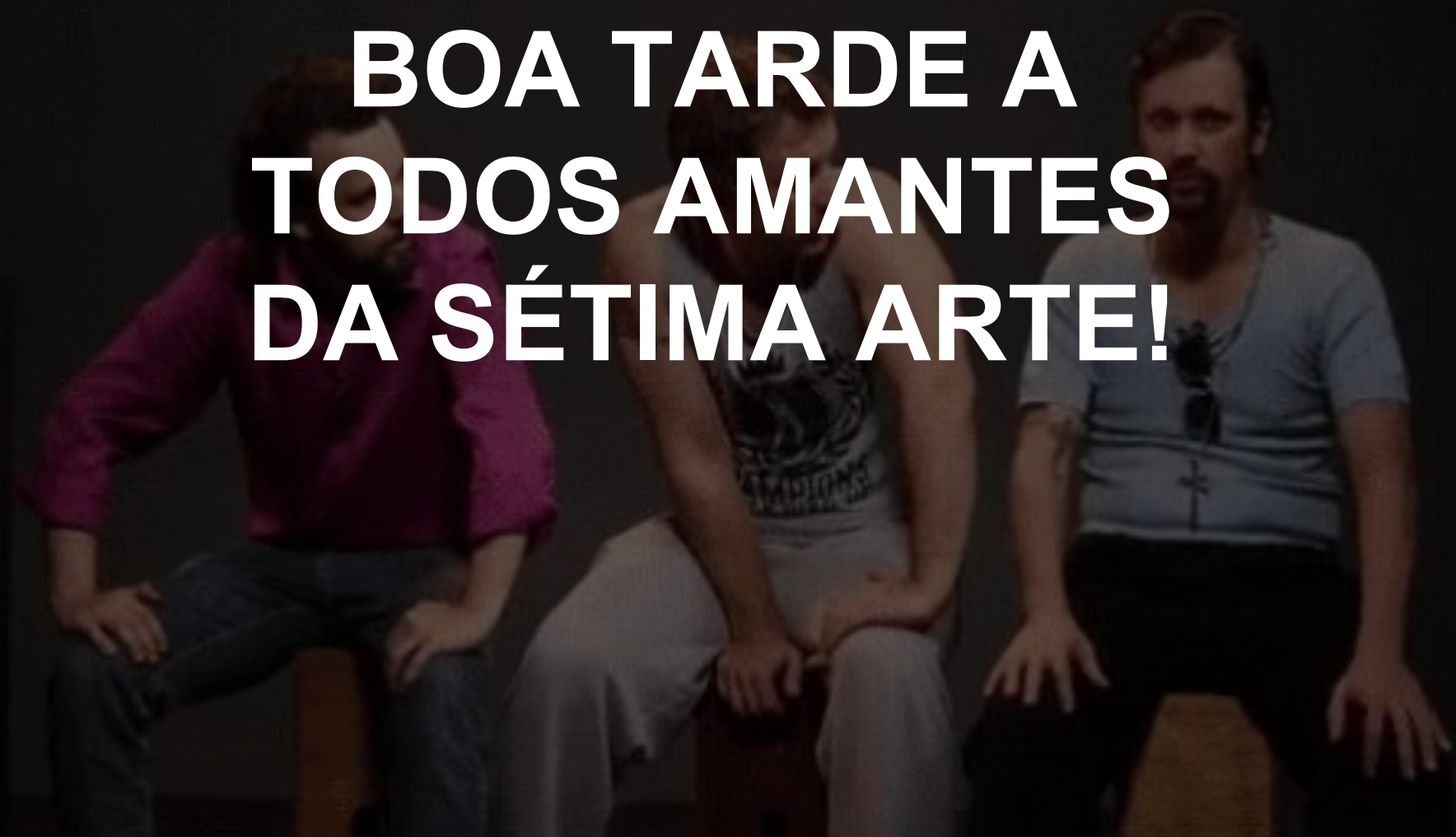
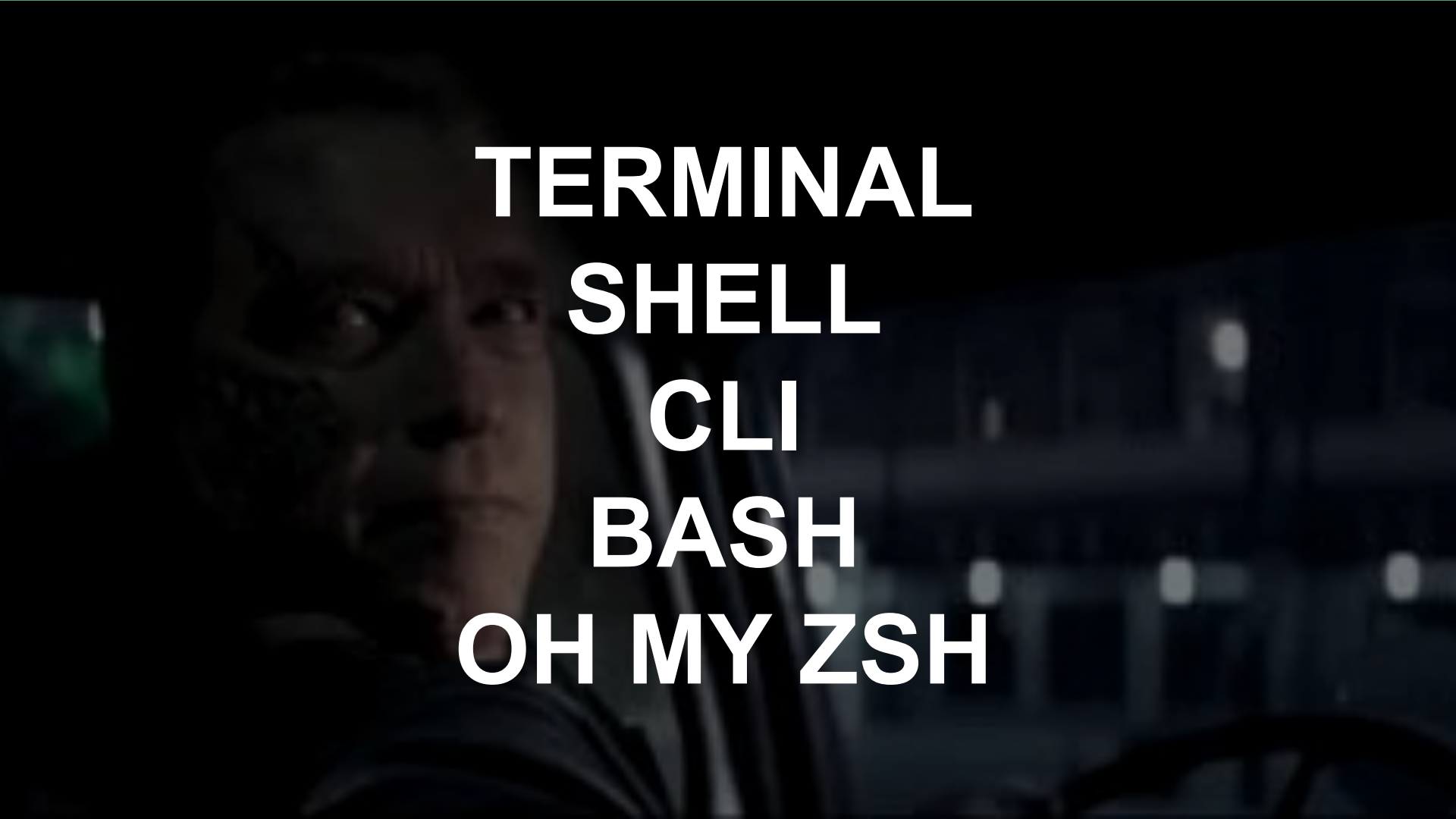


**BOA TARDE A  
TODOS AMANTES  
DA SÉTIMA ARTE!**

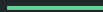




**TERMINAL  
SHELL  
CLI  
BASH  
OH MY ZSH**

# TERMINAL

- Ambiente para input/output de texto



# SHELL

- Uma shell (em português, casca) é uma interface de usuário para acessar os serviços de um sistema operacional.
- Utiliza de uma Command Line Interface (CLI) ao invés de uma Graphic User Interface (GUI)
- Interpretador de comandos de  
— linha

# BASH

- O bash é um interpretador de comandos, um entre os diversos tradutores entre o usuário e o sistema operacional conhecidos como shell.
  - Em suma, é um Shell do UNIX amplamente utilizado
-

# ZSH

- Uma alternativa ao BASH padrão do linux.
  - Um BASH bombadão ixiii
  - OH MY ZSH é uma framework para gerenciar o ZSH e melhorar o workflow
-

[projects]\$ rm -f foo.txt

prompt

command

argument

cursor

# Alguns Comandos CLI

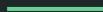
- ls
- cd
- touch
- rm
- mkdir
- mv





# Padrão de pastas


- Pasta corrente .
- Pasta pai ..
- Pasta raiz do usuário ~



# Leitura complementar

<https://www.learnenough.com/text-editor-tutorial>

---

A woman with blonde hair is sitting on a striped towel outdoors. She is holding a green apple in her right hand and looking towards the camera with a slight smile. The background is a dark, textured surface, possibly a wall or a large tree trunk. The text "TODO MUNDO DE BOA ATÉ AGORA???" is overlaid in white, bold, sans-serif font.

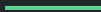
**TODO MUNDO DE  
BOA ATÉ  
AGORA??**

# EDITOR DE TEXTO



# EDITOR DE TEXTO

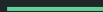
- Editor de texto vs Processador de palavras
- Editor de texto é uma ferramenta essencial para a computação moderna. Permite editar arquivos plain text.



# EXEMPLOS DE EDITORES DE TEXTO MODERNOS

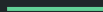
- SUBLIME
- ATOM
- VS CODE

DICA DE MESTRE: APRENDA  
TODOS OS ATALHOS!



# DICAS

- Aprenda os atalhos de teclado to editor, do navegador, do linux, de tudo
- Use o auto complete pelamor



# Leitura complementar

<https://www.learnenough.com/text-editor-tutorial>

---



# Curiosidade do dia





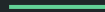
# **GIT & GITHUB**

GIT = A COISA  
MAIS LINDA  
QUE JÁ EXISTIU

- Jogue na cara das pessoas que não sabem git que elas não sabem git
  - Ore ao criador do linux e agradeça por tudo que ele já criou pra tu
-

# VERSION CONTROL SYSTEM

- VCS são uma categoria de ferramentas de software que auxiliam um time a controlar as modificações a um código ao longo do tempo



# VERSION CONTROL SYSTEM

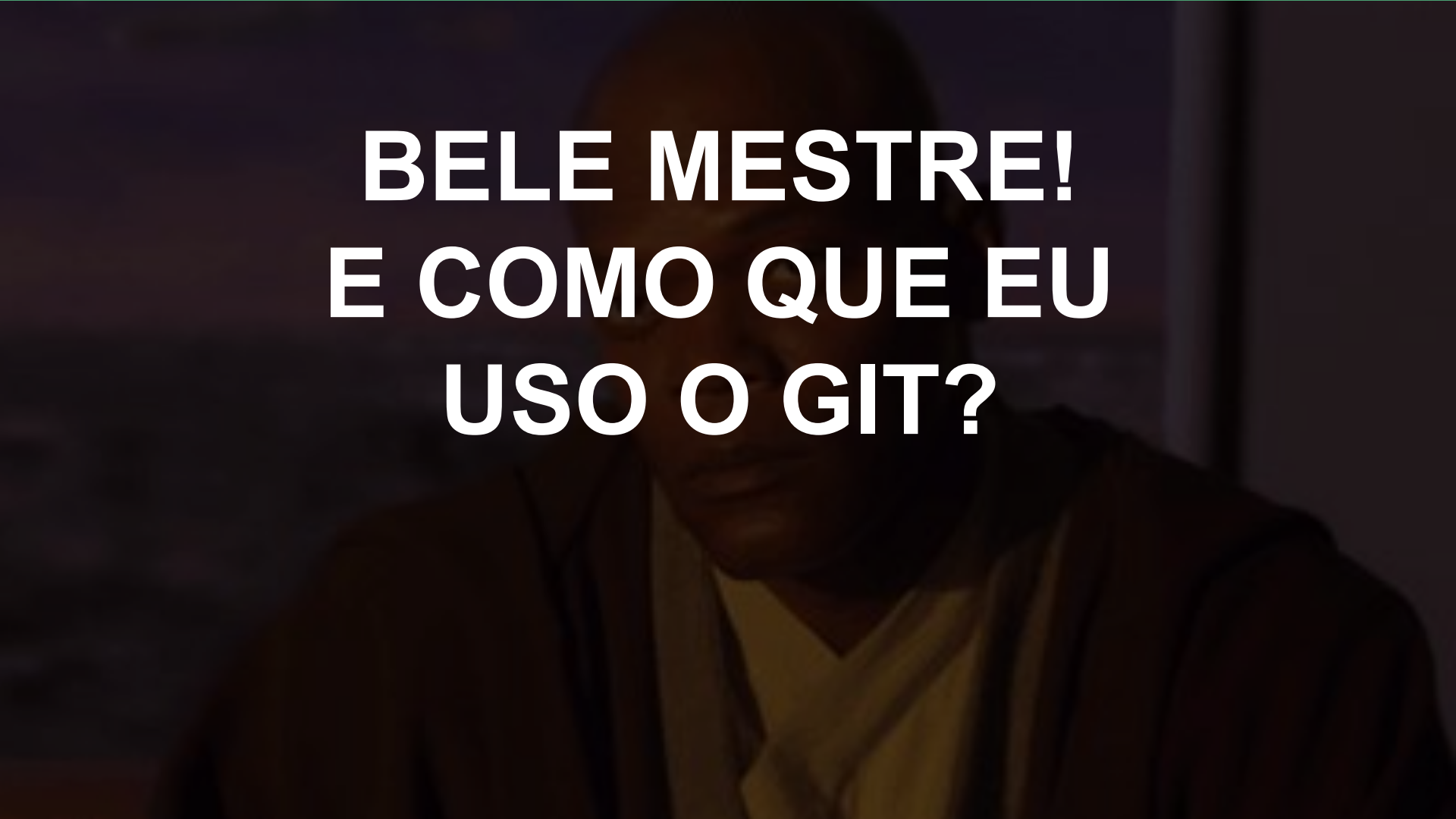
- Se um erro é cometido, uma versão antiga do código é resgatada
  - Permitem que diferentes membros trabalhem ao mesmo tempo em partes distintas do código
  - São ferramentas essenciais para abolir GoHorse, envio de código por email e outras blasfêmias
-

# GIT

- O VCS moderno mais difundido e usado atualmente
  - Desenvolvido originalmente pelo Linus (criador do Linux) em 2005
  - Open Source
-

# GITHUB

- GitHub é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git
  - Permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo
  - Uma rede social de programadores
-



**BELE MESTRE!  
E COMO QUE EU  
USO O GIT?**



# INICIANDO

- Todo projeto tem que começar de algum lugar...
- Se for localmente: `git init`
- Se o código já está disponível no github, clone-o: `git clone`

Você está criando um repositório local de ambos os jeitos

---

Seu repositório  
local contém 3  
árvores

- Working directory
  - Index ( Stage )
  - HEAD
-



working  
dir

add



Index  
(Stage)

commit



HEAD

# Comandos para mexer com stages e Head

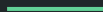
- `git add`
  - `git add goiaba.txt`
  - `git add -A`
  - `git add '*.txt'`
  - `git commit -m 'bla '`
  - `git log`
  - `git status`
  - `git diff`
-

# Enviando alterações

- Seu repositório local deve se conectar à um servidor
- Se você não clonou antes: `git remote add origin <servidor>`

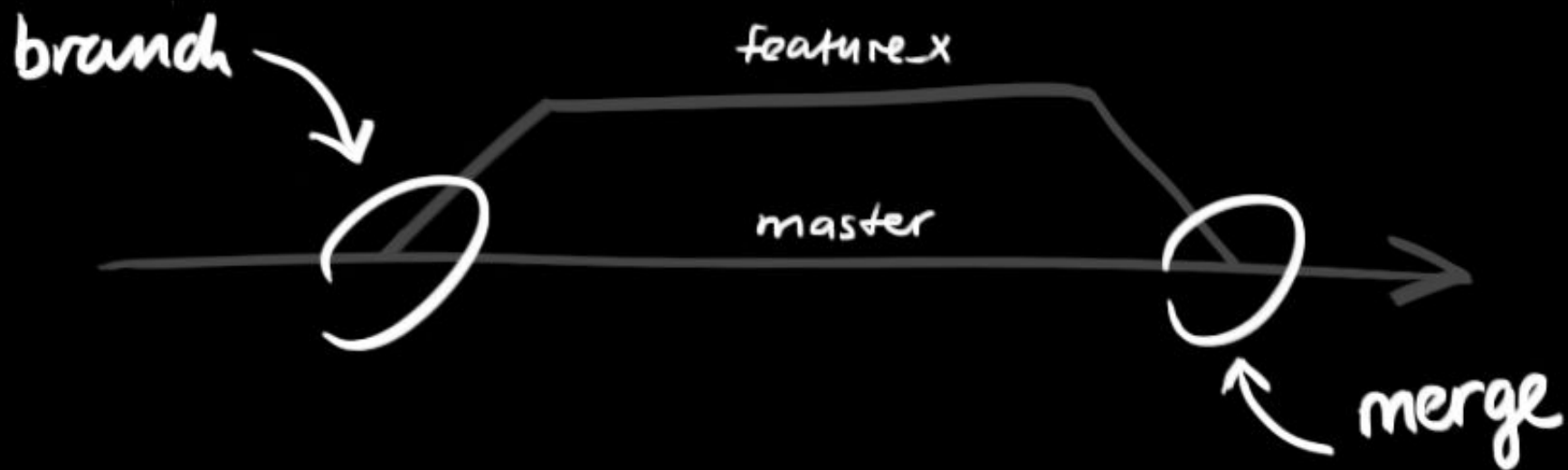
`git push origin master (ggpush)`

`git pull origin master (ggpull)`



# Ramificando

- Branches ("ramos") são utilizados para desenvolver funcionalidades isoladas umas das outras.
  - O branch master é o branch "padrão" quando você cria um repositório.
  - Use outros branches para desenvolver e mescle-os (merge) ao branch master após a conclusão.
-



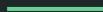
# Comandos usados para ramificações

- `git branch`
  - `git checkout`
  - `git checkout -b`
  - `git checkout -f`
  - `git merge <nome da branch>`
  - `git branch -d`
-



# Rapidamente sobre o padrão de versionamento utilizado na CJR

- master
- develop
- featureX



# Leitura Complementar

- <https://www.atlassian.com/git/tutorials/what-is-version-control>
  - <https://try.github.io/levels/1/challenges/1>
  - [http://rogerdudler.github.io/git-guide/index.pt\\_BR.html](http://rogerdudler.github.io/git-guide/index.pt_BR.html)
-



# **GIT MERGE**

## **CABO A AULA**