

Introdução a Gerência de configuração

Programa de formação em
Dados - Lojas renner SA





Lembrar de Gravar

INSTRUTOR

Eduardo Maia

Engenheiro SW no CESAR

Mestrando em Sistemas Distribuídos

Estatística e Tratamento de dados



ehammo@cesar.org.br



Monitores

Sou um aluno do 3º Período da CESAR SCHOOL, atualmente estagiando no CESAR, e trabalhando com C# e AzureDevOps. Faço parte de uma startup chamada Petech que procura solucionar dores de pais e mães de pets.

Sou uma pessoa bem descontraída e serena, apaixonado por estudar a área de psicologia. Sou completamente pirado na solução de problemas e conversas sobre diferentes perspectivas de soluções, entrei na área de computação justamente por essa paixão ardente em resolver problemas martelando a cabeça na parede e discutindo com pessoas ao longo do caminho. Além disso, tenho um Pug chamado Max que só come e dorme.

Lucas Borges



Monitores

Iniciei minha jornada na área de tecnologia após terminar uma graduação em Direito, através de estudos autodidatas, e atualmente sou acadêmico de Ciência da Computação no 3º período. Já tentei adentrar em algumas outras áreas, como a de tradução, pois gosto muito da área de línguas, e modelagem 3D para efeitos visuais algo que sempre fui fascinado e curioso sobre como funcionava. Porém, após minha primeira imersão na programação, me apaixonei e decidi trabalhar na área.

Nas poucas horas vagas que tenho, tento fazer coisas como assistir séries e filmes, fazer exercícios e passar tempo com a minha cadelinha chamada Aylla.

Lucas Fernandes



Monitores

Sou estudante de Ciência da Computação na CESAR School e atualmente estou no 3º período. Sou apaixonado tecnologia desde de pequeno e está sendo um caminho incrível, já fui monitor algumas vezes e estou muito feliz de ser monitor nesse curso e está ajudando vocês nesse processo. Sou uma pessoa extrovertida, gosto de conhecer pessoas novas e sou muito Cinéfilo, sério posso conversar sobre filmes e séries por dias.



Pedro Henrique



CRONOGRAMA

- **1. Introdução**

Por que usar VCS? O que é Git? O que é Github? Integração com Colab

- **2. Conceitos Básicos de git**

Comandos básicos de Git: Add, Commit, Push, Pull, Status, log, diff, checkout, reset;

- **3. Além do Básico**

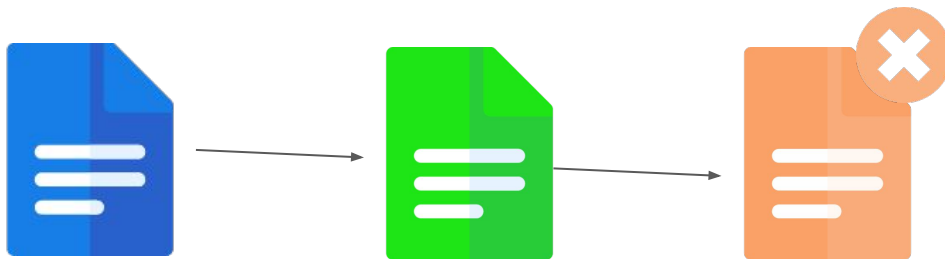
Branches, conflitos, merge, rebase



Introdução

O que é/Por que usar VCS?

- VCS é Version Control System, ou sistema de controle de versão



O que é/Por que usar VCS?

- VCS é Version Control System, ou sistema de controle de versão



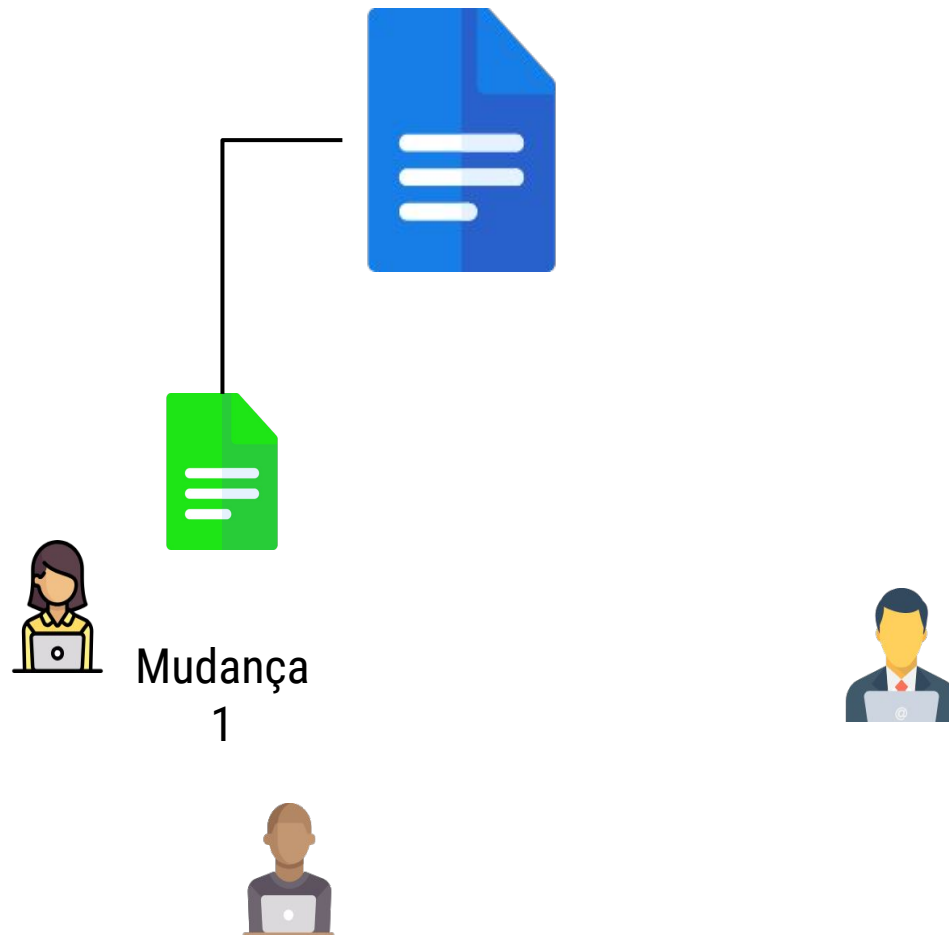
Final

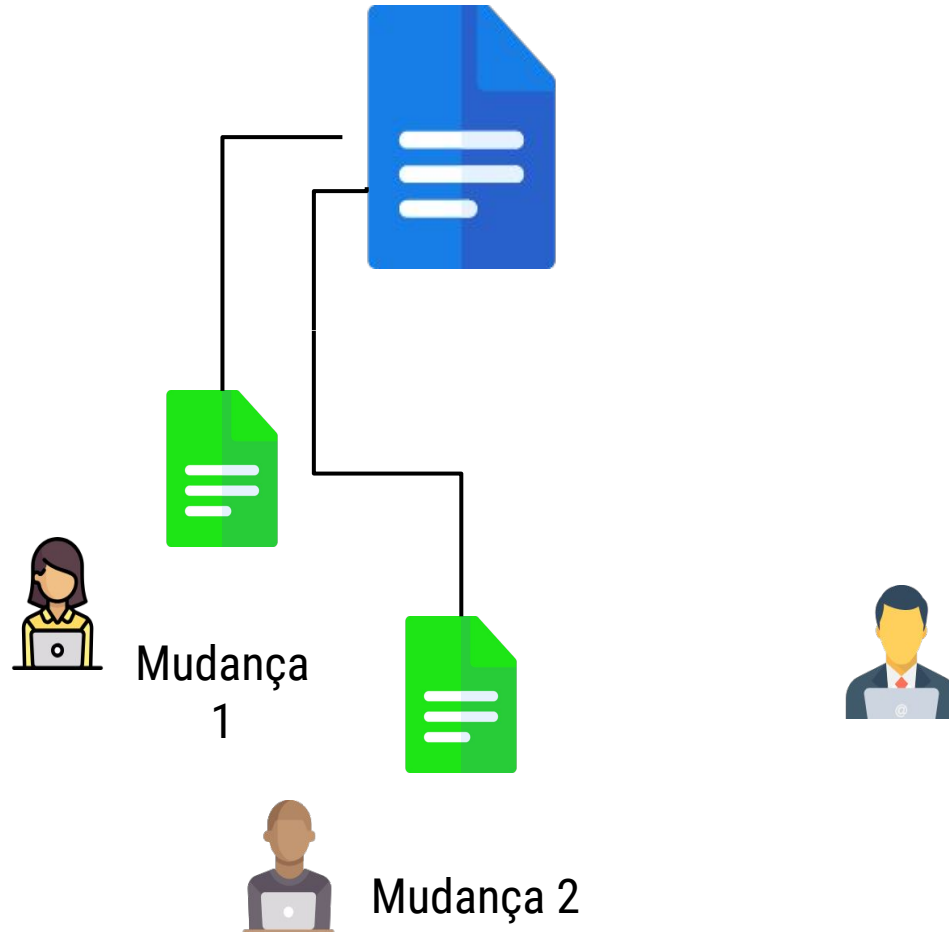


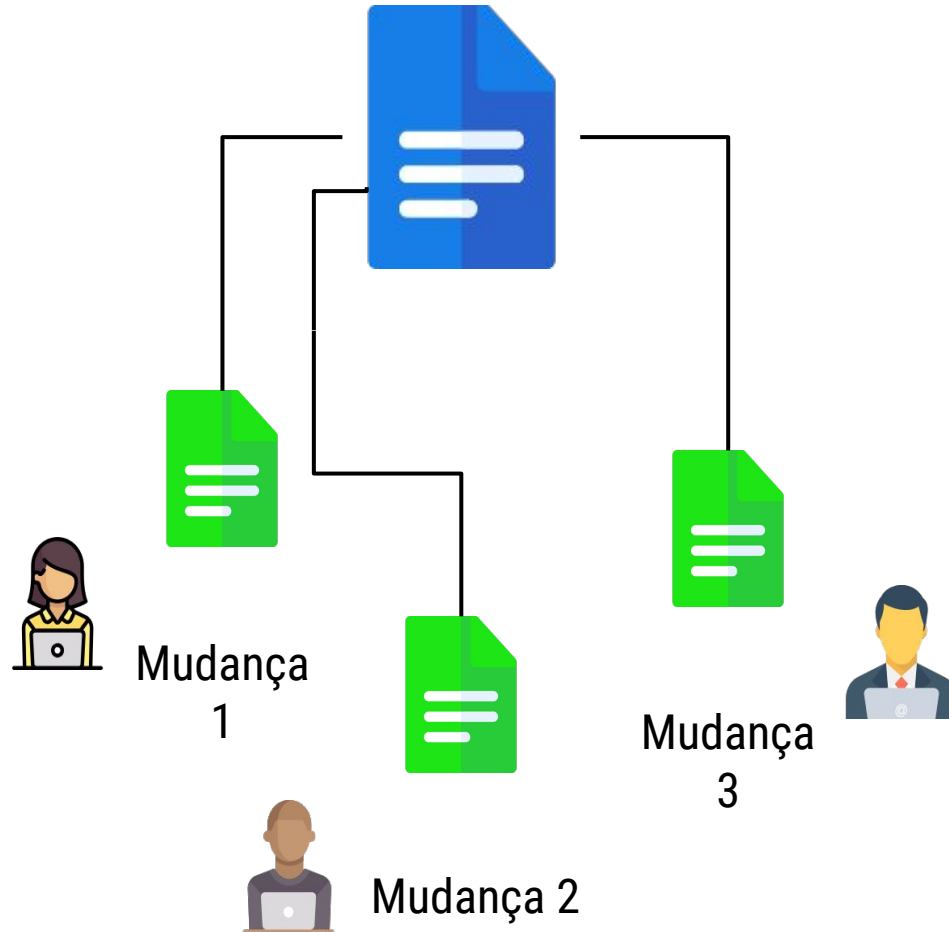
Final_Final

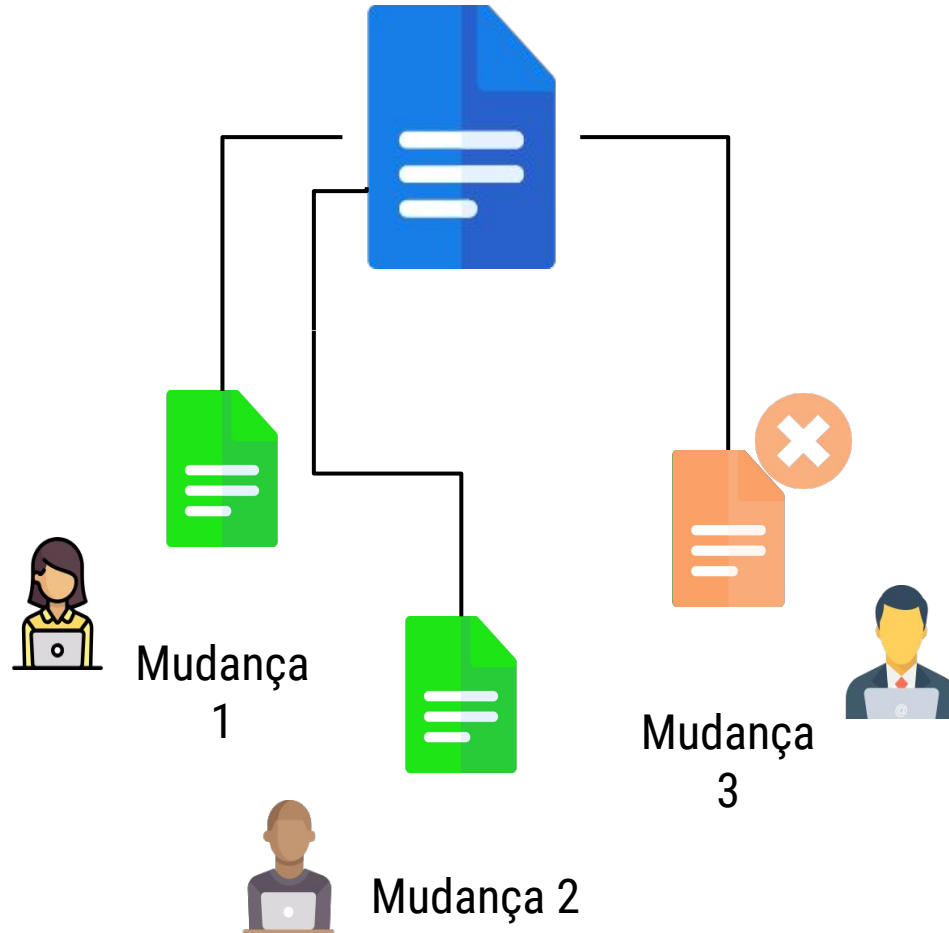


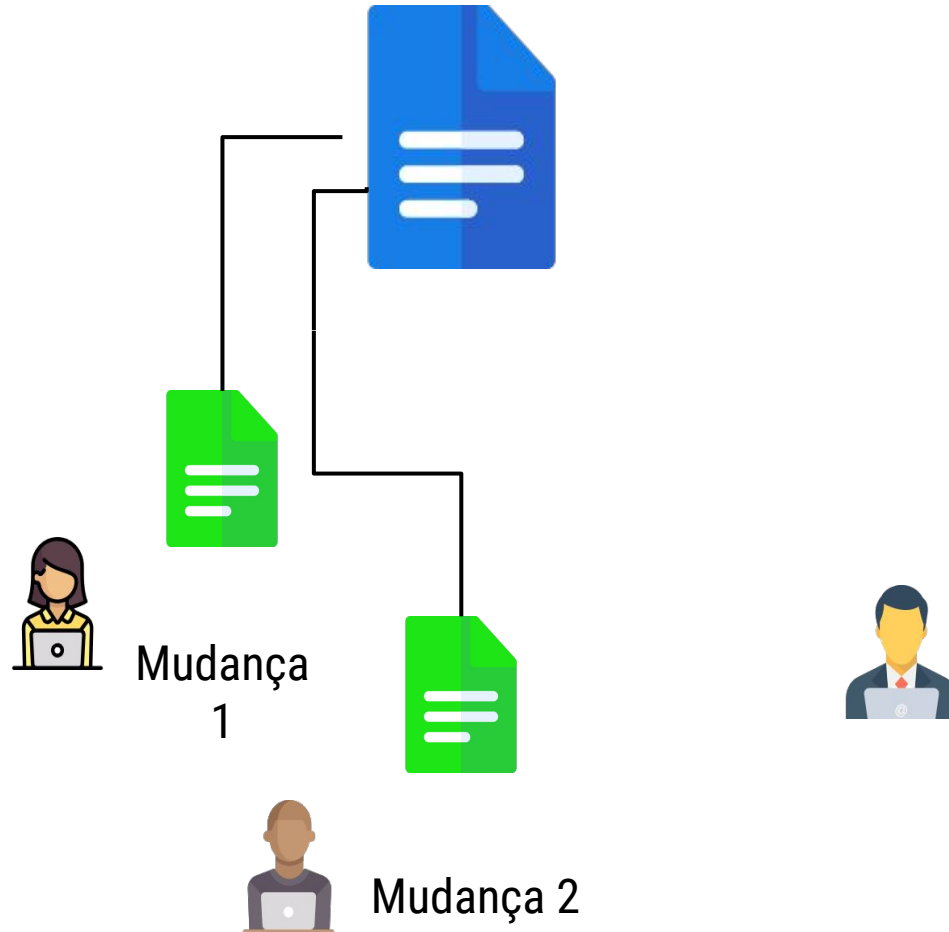
Final_agora_vai

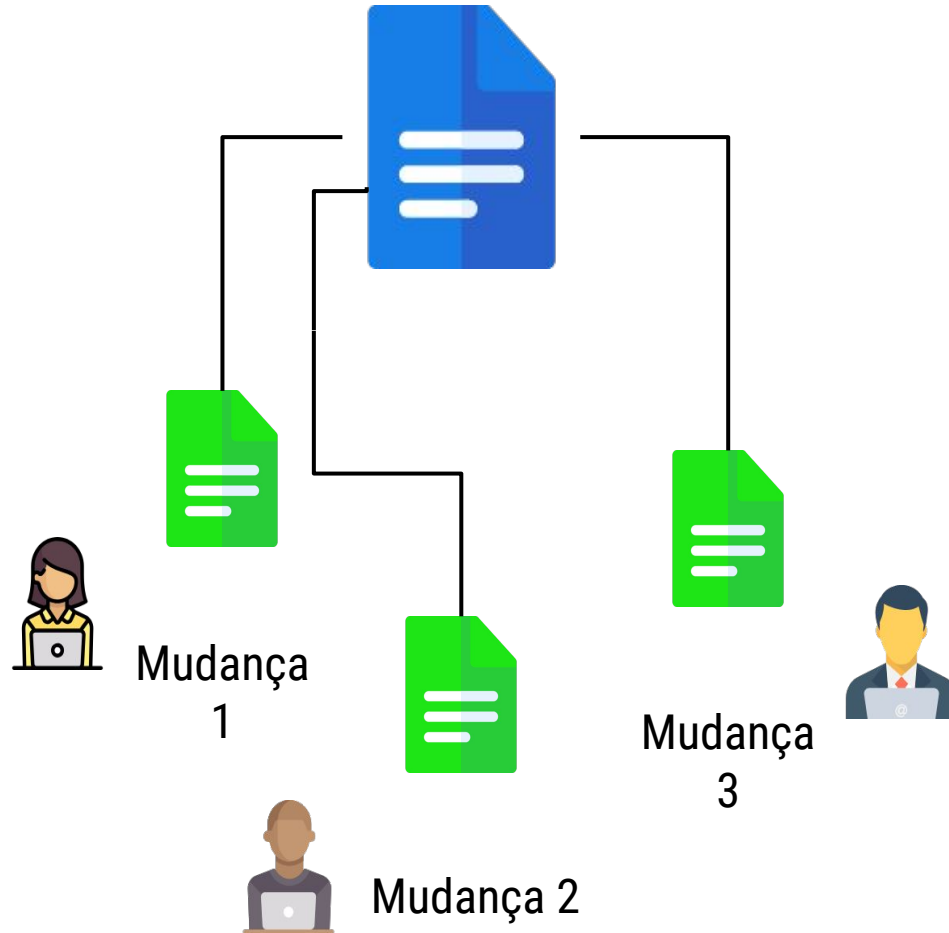












O que é Git?

- É um sistema de controle de versão
- Concorrentes: SVN, CSV
- Design Simples e robusto
- Diferencial: Distribuído

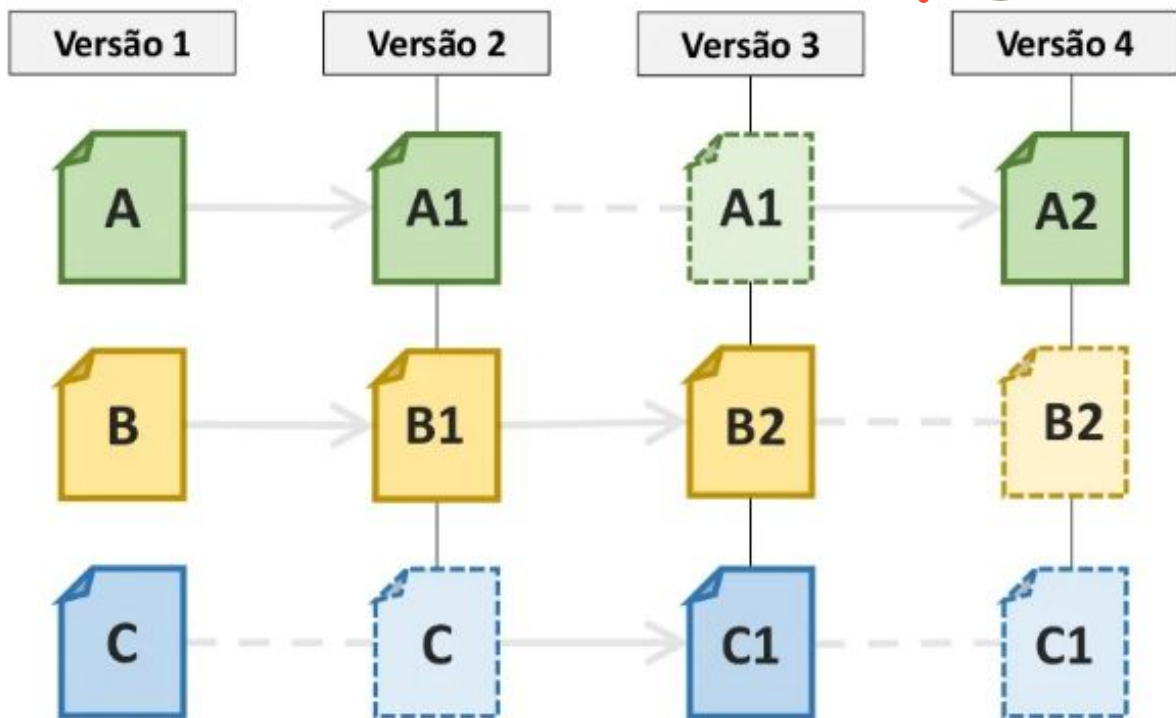




O que é Git?

- Download: <https://git-scm.com/downloads>

Sistema de controle do git





O que é Github?

- Um servidor na Nuvem que hospeda repositórios Git
- Uma rede social de desenvolvedores
- A “casa” da maioria dos projetos Open Source
- Portfólio de desenvolvedores

Git Vs Github

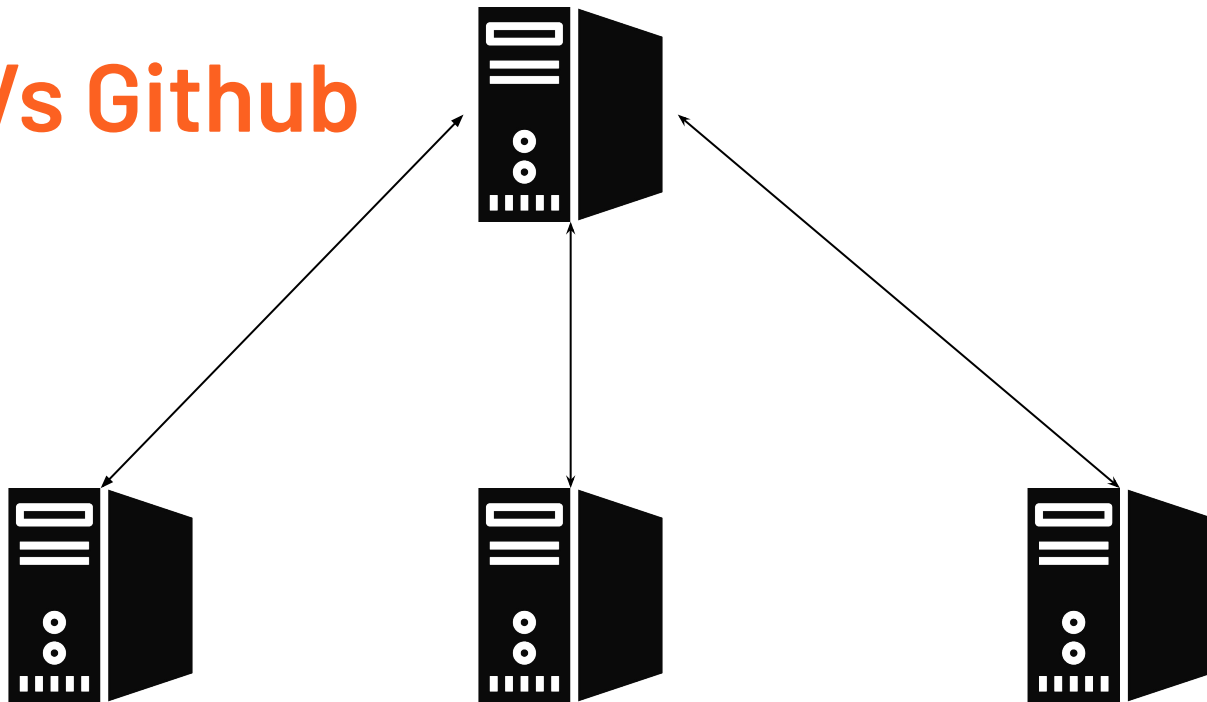


git

!=



Git Vs Github



Mac de Pedro

Ubuntu de Eduardo

Windows de Lucas

Github Concorrentes





Github

- Criem uma conta: <https://github.com>
- Criem um repositório público chamado Git-Renner-Aula01 já com um README

Github



Repositories



New

Find a repository...

FASE 1

FASE 2

FASE 3

FASE 4

Github

Owner *



ehammo ▾

Repository name *

Git-Renner-Aula01 ✓

Great repository names

Git-Renner-Aula01 is available. Need inspiration? H

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can co



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.



Add a README file

This is where you can write a long description for your project. [Learn more.](#)

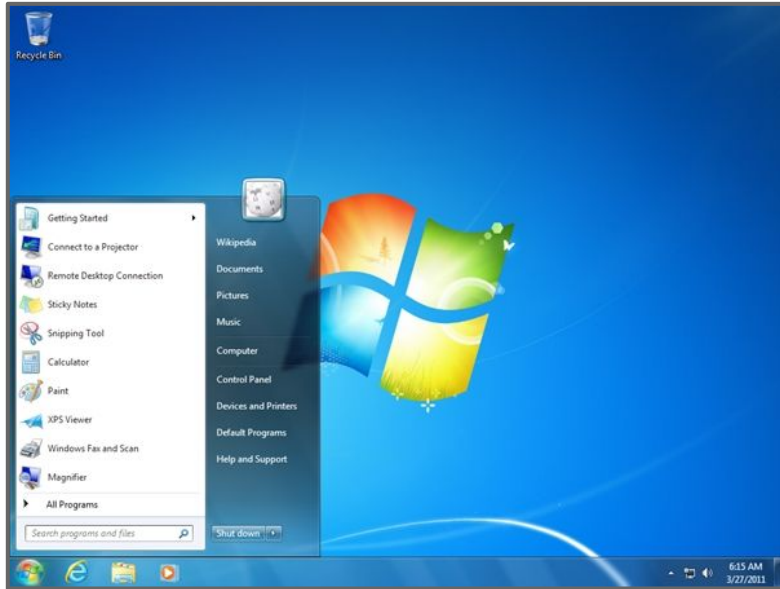


Github no Colab

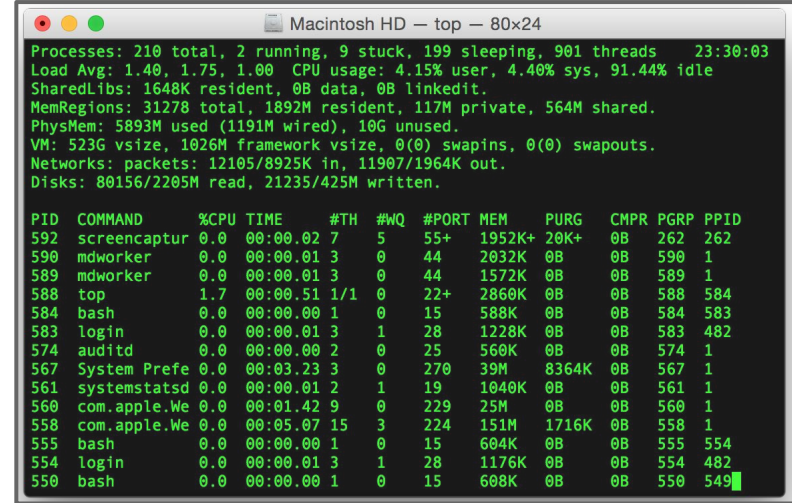


CLI: Command Line Interface

Tipos de interface



GUI - Graphical user interface

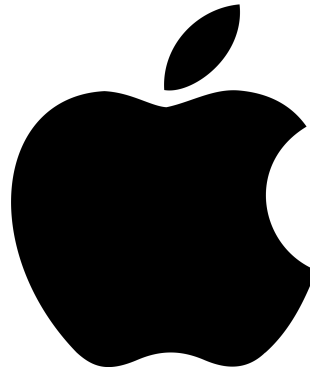
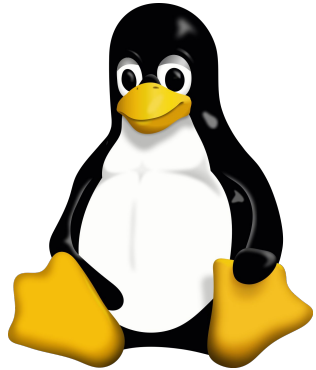


CLI - Command Line Interface

Quando usar CLI's?

- Acesso remoto a servidores
- Automação de tarefas/criação de scripts
- Maior controle e poder sobre os comandos executados
- Com o tempo se torna mais rápido e eficiente que usar UI

Unix Vs MS-Dos

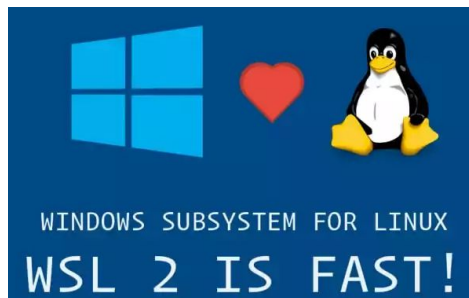


**Sistemas baseados
em Unix**



Windows (MS-DOS)

Eu uso Windows e agora?





Lembrar de Gravar

Como usar CLI?

- `mkdir`: Cria uma pasta
- `ls`: Lista arquivos e pastas
 - `ls -a` (mostra pastas e arquivos ocultos)
 - `ls -la` (Igual a `ls -a` exibindo permissões e ownership)
- `cd {caminho}`: Mudar para a pasta em {caminho}
 - `cd ..` (.. se referência a pasta anterior)
 - `cd /` (Diretorio raiz, ou `c://` no windows)
 - `cd ~` (Home, ou `C:/Users/Usuario` no windows)
 - `cd -` (Volta para a ultima pasta que você estava)

Como usar CLI?

- `pwd`: Imprime a pasta que estamos
- `rm {-rf} {arquivo/pasta}`: Remove arquivo ou pasta
- `clear`: Limpa o terminal
- `cat arquivo`: Imprime o conteúdo do arquivo
- `vim arquivo`: Abre/Cria um arquivo no vim
- `shift+insert` ou botão direito>`paste`: Cola o que estiver em sua área de transferência
- `<tab>`: Auto completa nome de arquivos e pastas

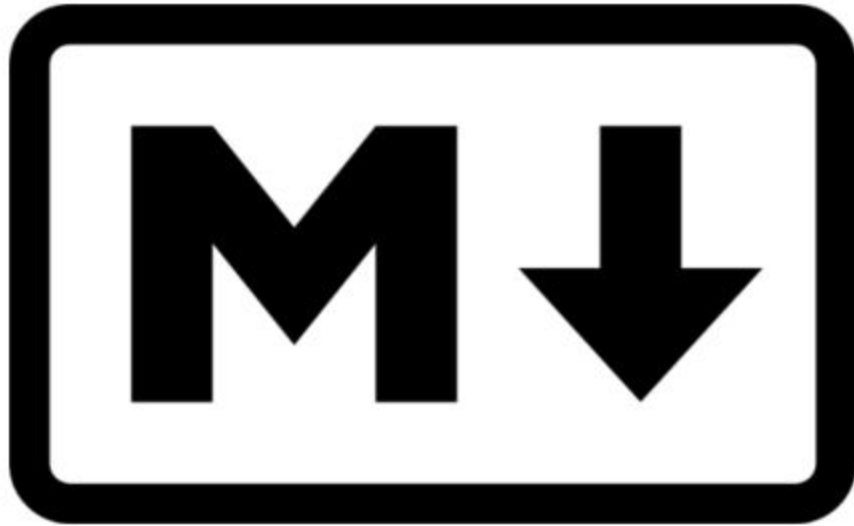
Dá para programar só na linha de comando?



Como usar VIM?

- No modo normal:
 - i -> Entra no modo insert
 - dd -> Deleta uma linha
 - GG -> Vai para o final do arquivo
 - :wq -> Sai do arquivo
 - /palavra -> busca palavra no arquivo
- No modo de inserção
 - <esc> -> Volta pro modo normal

Markdown



HEADERS

```
# This is an <h1> tag
## This is an <h2> tag
##### This is an <h6> tag
```

EMPHASIS

```
*This text will be italic*
_This will also be italic_

**This text will be bold**
__This will also be bold__

*You **can** combine them*
```

BLOCKQUOTES

As Grace Hopper said:

```
> I've always been more interested
> in the future than in the past.
```

As Grace Hopper said:

```
| I've always been more interested
| in the future than in the past.
```

LISTS

Unordered

- * Item 1
- * Item 2
 - * Item 2a
 - * Item 2b

Ordered

1. Item 1
2. Item 2
3. Item 3
 - * Item 3a
 - * Item 3b

IMAGES

```
![GitHub Logo](/images/logo.png)
```

Format: `![Alt Text](url)`

LINKS

```
http://github.com - automatic!
```

```
[GitHub](http://github.com)
```

BACKSLASH ESCAPES

Markdown allows you to use backslash escapes to generate literal characters which would otherwise have special meaning in Markdown's formatting syntax.

```
\*literal asterisks\*
```

```
*literal asterisks*
```

Markdown provides backslash escapes for the following characters:

\ backslash	() parentheses
` backtick	# hash mark
* asterisk	+ plus sign
_ underscore	- minus sign (hyphen)
{ curly braces	. dot
[] square brackets	! exclamation mark



Comandos Básicos de git



Comandos Básicos de git

Comandos básicos de git

- Git init

```
ehammo@PERDIZES-016228:~/Documents/school/projeto-aula02$ git init
Initialized empty Git repository in /home/ehammo/Documents/school/projeto-aula02/.git/
ehammo@PERDIZES-016228:~/Documents/school/projeto-aula02$ ls -a
.  ..  .git
```

Comandos básicos de git

- Configurando seu Email e Nome:
 - `git config --global user.email "ehammo@cesar.school"`
 - `git config --global user.name "Eduardo Maia"`
 - `git config --list`
 - `cat ~/.gitconfig`

Exercício

- Crie um repositório com git init
- Usando vim crie:
 - Um arquivo README.md sobre você em Markdown
 - 3 arquivos vazios chamados index.html, styles.css, scripts.js
- Criem um commit contendo apenas os arquivos vazios

```
# Página pessoal de Fulano
```

```
Olá, eu sou Fulano, entrei nesse curso com tal objetivo. Sou da área X,  
estou migrando de carreira/me especializando em dados por conta de tal razão.
```

```
## Curiosidades sobre mim
```

- Amo cozinhar
- Odeio lavar a louça
- Tenho um cachorrinho

Diretório



index.html



styles.css



scripts.js



README.md

Staging

Repositório

Como selecionar arquivos para enviar à zona de staging?

Como selecionar arquivos para enviar à zona de staging?

- `git add arquivo`

Diretório



index.html



styles.css



scripts.js



README.md

Staging



index.html



styles.css



scripts.js

Repositório

Git Status

Diretório



index.html



styles.css



scripts.js



README.md

Staging



index.html



styles.css



scripts.js

Repositório

Como gerar nova versão dos arquivos que estão em staging?

Como gerar nova versão dos arquivos que estão em staging?

- `git commit`

Diretório



index.html



styles.css



scripts.js



README.md

Staging

Repositório



index.html



styles.css



scripts.js

Versionamento

Commit 1

Commit 2

Commit 3



Dicas para mensagem de commit

- Imperativo
- Evite o genérico: “Corrigir erros”, “Ajuste”
- O título deve expressar qual foi a mudança
- O conteúdo do commit deve expressar o motivo da mudança ter sido feita naquela forma
- Comece com letra maiúscula e não termine com ponto final
- Primeira linha: 50 char, cada linha de conteúdo 72 char
- Seja direto e objetivo! Nada de: Eu acho, talvez, tipo, ...

Exercício

- Gere um commit com o README
- Apague os arquivos vazios e gere um commit os removendo
- altere o README Acrescentando uma área que informa desde quando você programa e que linguagem você começou
- Gere um 3 commit alterando o README