

FOCAS

# Git & GitHub

Semana Acadêmica - 2025.2

# **Aviso!**

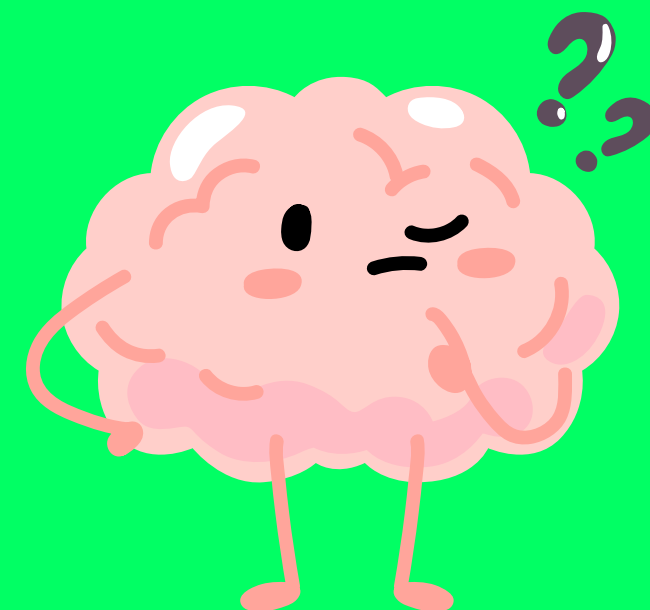
---

- Temos aqui pessoas de diversos períodos do curso.

Quaisquer questionamentos ou conceitos que não estejam familiarizados, avisem e tirarei as dúvidas que surgirem.



**Alguém aí já teve  
experiência com  
versionamento de código?**



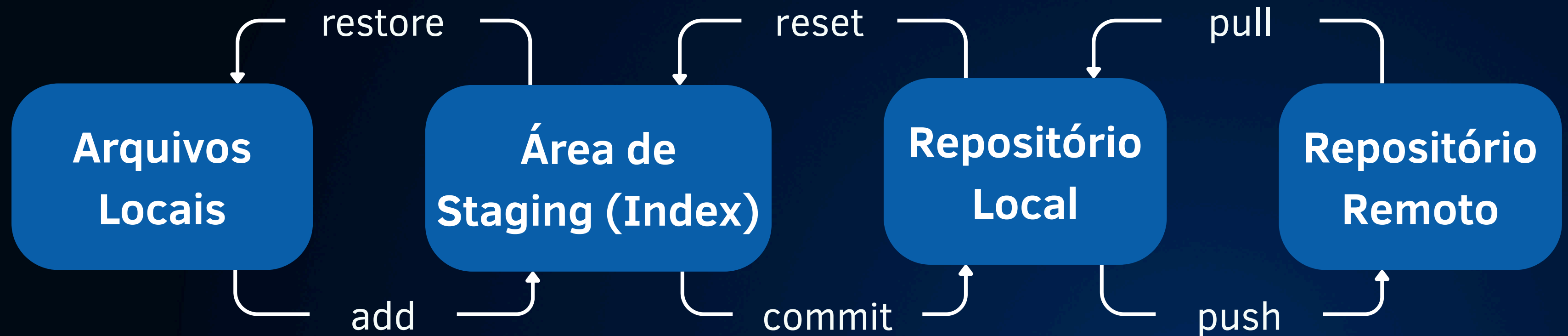
# Git

---

- Criado por Linus Torvalds, após problemas com a ferramenta de versionamento utilizada no projeto Linux
- Controla e salva as modificações feitas em arquivos
- Permite o desenvolvimento paralelo em equipe
- Amplamente adotado no mundo da computação



# Visão Simplificada



# Principais Comandos

---

- **git init** - inicia um repositório local
- **git add** - adiciona arquivos na área de staging
- **git commit** - “comprometer-se”, registra o estado atual dos arquivos no repositório local
- **git push** - envia o último estado salvo dos arquivos para o repositório remoto

```
brenopinna@BrenoHP:~/teste_git $ git init
Initialized empty Git repository in /home/brenopinna/teste_git/.git/
brenopinna@BrenoHP:~/teste_git > git(main) $ echo "testando o git!" >> file.txt
brenopinna@BrenoHP:~/teste_git > git(main) $ git add .
brenopinna@BrenoHP:~/teste_git > git(main) $ git status

On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file.txt

brenopinna@BrenoHP:~/teste_git > git(main) $ git commit -m "meu primeiro commit!"
[main (root-commit) a8c4339] meu primeiro commit!
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt
brenopinna@BrenoHP:~/teste_git > git(main) $ git status

On branch main
nothing to commit, working tree clean
brenopinna@BrenoHP:~/teste_git > git(main) $ █
```

# Conventional Commits

---

Várias pessoas terão acesso ao histórico de commits de um repositório. É essencial que o nome de um commit indique de forma clara e concisa o que foi feito nele, para melhor compreensão.



# Conventional Commits

---

- **Padrão de nomes**
  - `[emoji:] tipo: breve descrição`
- **Alguns tipos de commit:**
  - **feat** - novas funcionalidades
  - **fix** - correção de bugs
  - **refactor** - refatoração, sem mudanças na funcionalidade

# Conventional Commits

---

```
brenopinna@BrenoHP:~/teste_git > git(main) $ echo "mudei!" >> file.txt
brenopinna@BrenoHP:~/teste_git > git(main) $ git add .
brenopinna@BrenoHP:~/teste_git > git(main) $ git commit -m ":sparkles: feat:
adiciona uma otima funcionalidade"
[main 2b2ec8d] :sparkles: feat: adiciona uma otima funcionalidade
1 file changed, 1 insertion(+)
brenopinna@BrenoHP:~/teste_git > git(main) $ git log --oneline
2b2ec8d (HEAD -> main) :sparkles: feat: adiciona uma otima funcionalidade
a8c4339 meu primeiro commit!
brenopinna@BrenoHP:~/teste_git > git(main) $ █
```

# Branches

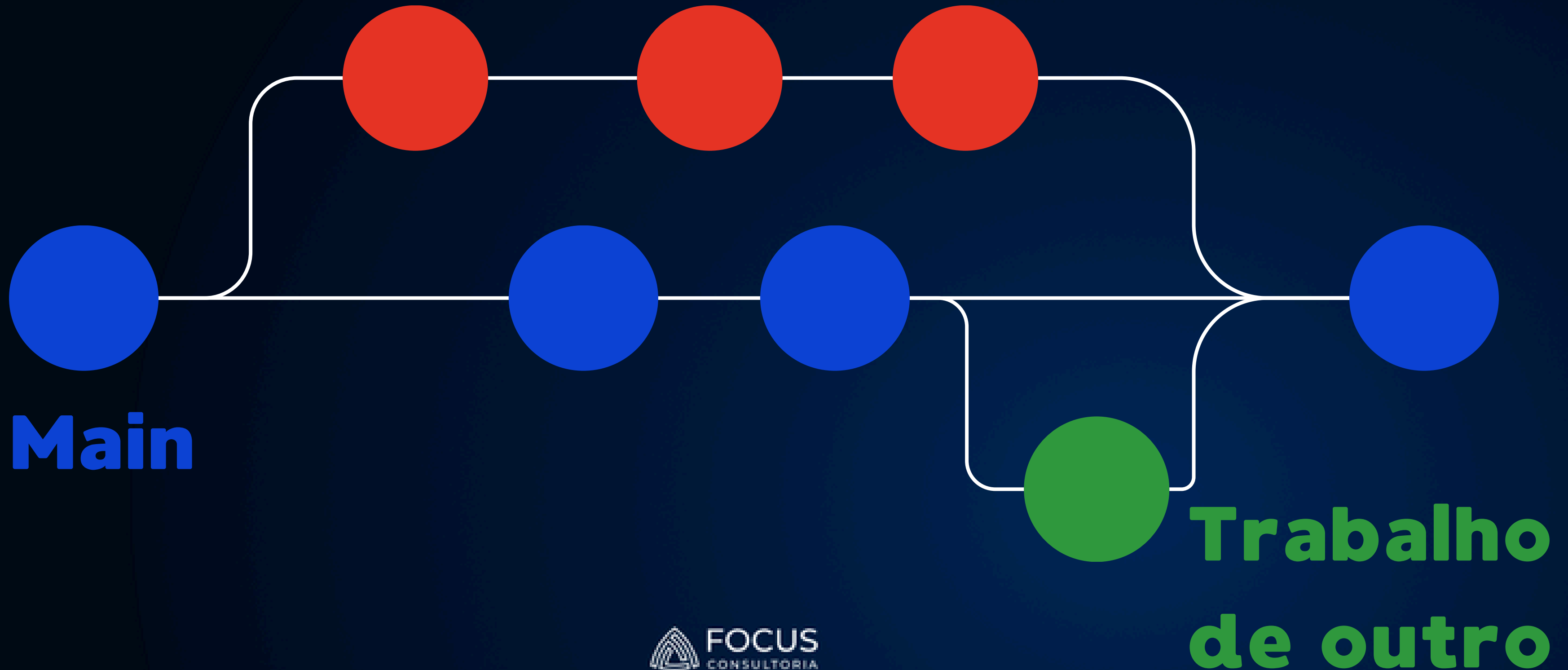
---

O Git permite trabalho paralelo, o que significa que mais de uma pessoa pode trabalhar no código ao mesmo tempo.

Para isso, existem as **branches** (ramificações)

# Branches

**Seu trabalho**



“Mas como vou trabalhar em  
equipe se está tudo na minha  
máquina?”

# GitHub

---

- Ferramenta para trabalhar colaborativamente em códigos
- Uma das mais utilizadas na atualidade
- Pode servir como seu portfólio de projetos
- Integração com repositórios Git



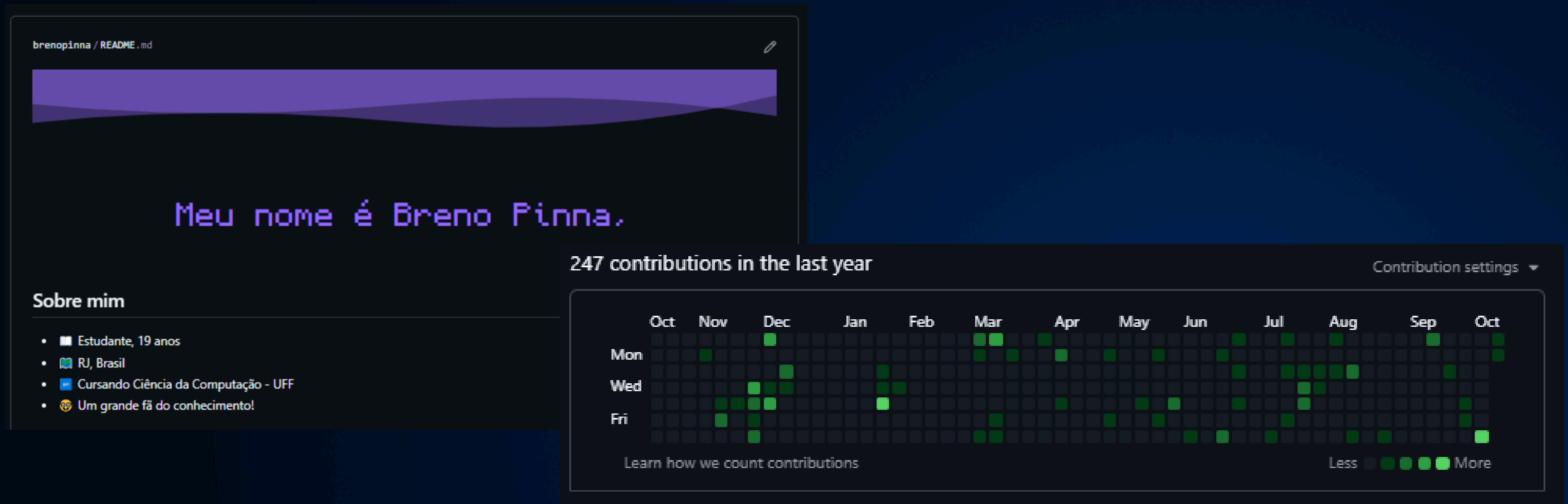
# Principais Conceitos

---

- **README:** Um arquivo em Markdown, geralmente com instruções de instalação e execução sobre um código fonte...

# Principais Conceitos

- ...ou até mesmo com a apresentação do dono da conta.





# Principais Conceitos

---

- **Fork:** “Bifurcação”, consiste em criar um ramo em um repositório de terceiros, para que depois contribua com suas alterações.
- **Pull Request:** uma solicitação para que seu código entre no repositório de um terceiro

# **Linha de Comando: Inimiga ou Aliada?**

---

- Com a modernidade das interfaces simplificadas e intuitivas, muitos temem a linha de comando dos computadores
- Pensando nisso, decidiram tornar o dia do estudante de Git um pouco mais fácil...

# Download GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

[Download for Windows \(64bit\)](#)

## Try beta features and help improve future releases

Experience the latest features and bug fixes before they're released.

[Check out Beta](#)

## Prefer the MSI?

Need to download the package to install across your organization?

[Download for Windows \(MSI\)](#)

## Mac?

Need to download for macOS?

[Download for macOS](#)

# **Linha de Comando: Inimiga ou Aliada?**

---

- Não fuja do terminal, ele vai te encontrar mais cedo ou mais tarde.
- Mas enquanto aprende, use essa ferramenta se achar melhor
- Não está disponível oficialmente para Linux (pare de fugir e aprenda a usar a linha de comando!)

# Prática: Criando um repositório

---

Agora, vamos unir todos esses conceitos e criar nosso primeiro repositório git, e sincronizá-lo com o GitHub!

# Alguma pergunta?