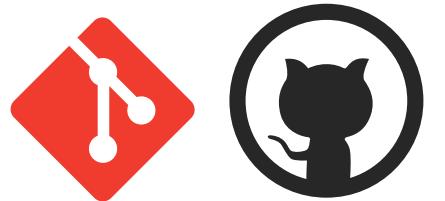


September 14, 2024

8:00 AM – 12:00 PM



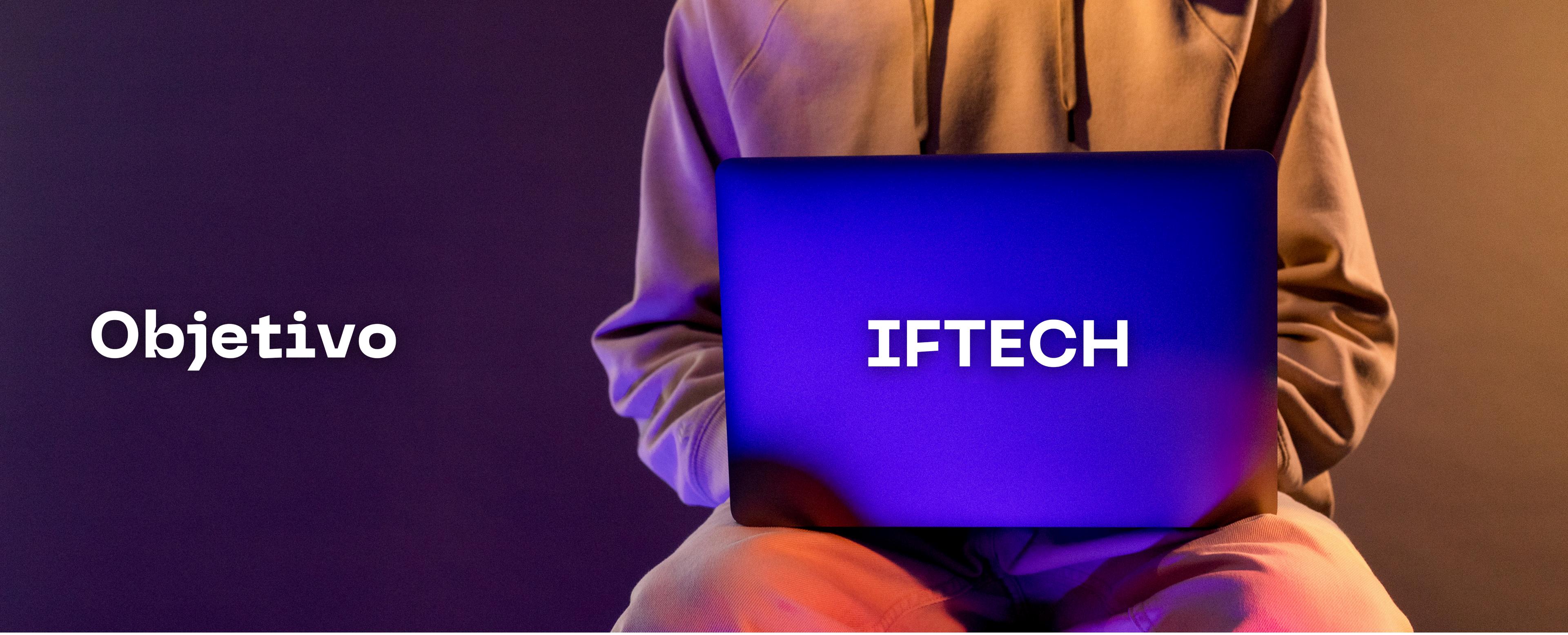
# Git e Github: guia básico de sobrevivência

📍 IFPB – Campus João Pessoa

**By: Leticia e Suetone**



# Objetivo



IFTECH

Entender o que é Git e GitHub, o que é versionamento e sua importância. Aprender os comandos básicos para qualquer programador sobreviver no mercado.



# Tópicos



Git



GitHub



01

## Introdução

Versionamento, repositórios e a diferença entre Git e Github

02

## Comandos básicos

Básicos, porém importantes!

03

## Prática

Colocando em prática o que aprendemos :)

04

## Como usar o chatGPT a seu favor?

Essa é bônus...

# **Resumo e links úteis:**

[https://github.com/leticiazlopes/  
guia-basico-git](https://github.com/leticiazlopes/guia-basico-git)



# Pense no seguinte cenário:



Você está trabalhando em um projeto de gestão para o IFTECH e faz parte de uma equipe com mais 8 devs.

O projeto vai ter frontend, autenticação, consumo de APIs, cadastro de atividades e usuários, entre outras funcionalidades...

# Projeto de gestão do IFTECH

Como gerenciar o trabalho dessas pessoas de maneira eficiente?



projeto\_iftech\_tela\_login.js

projeto\_iftech\_banco\_de\_dados.db

projeto\_iftech\_admin.py

projeto\_iftech\_versao1.zip

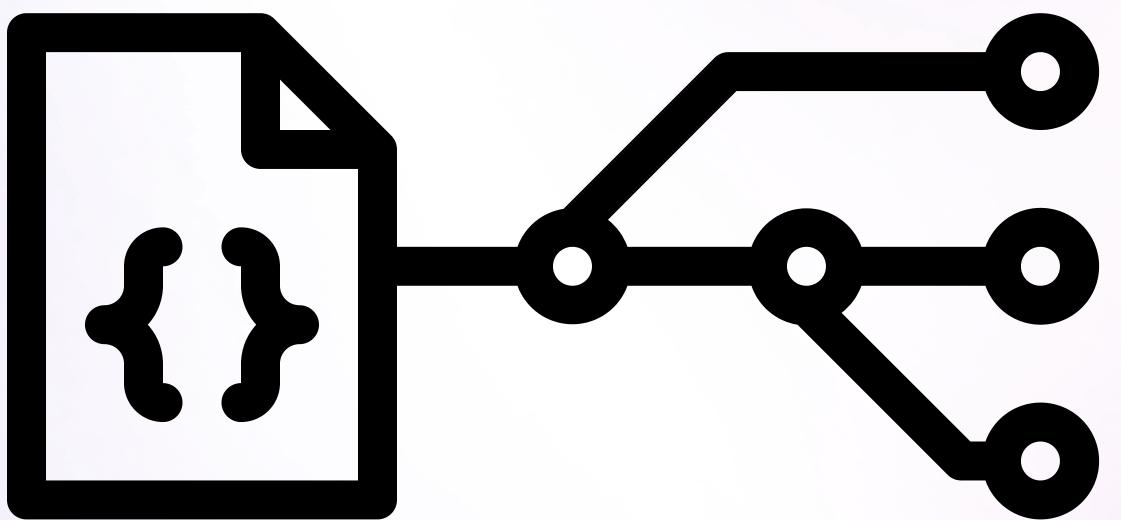
projeto\_iftech\_versao1.1.zip

projeto\_iftech\_versao\_final.zip

projeto\_iftech\_versao\_final\_sem\_bug.zip

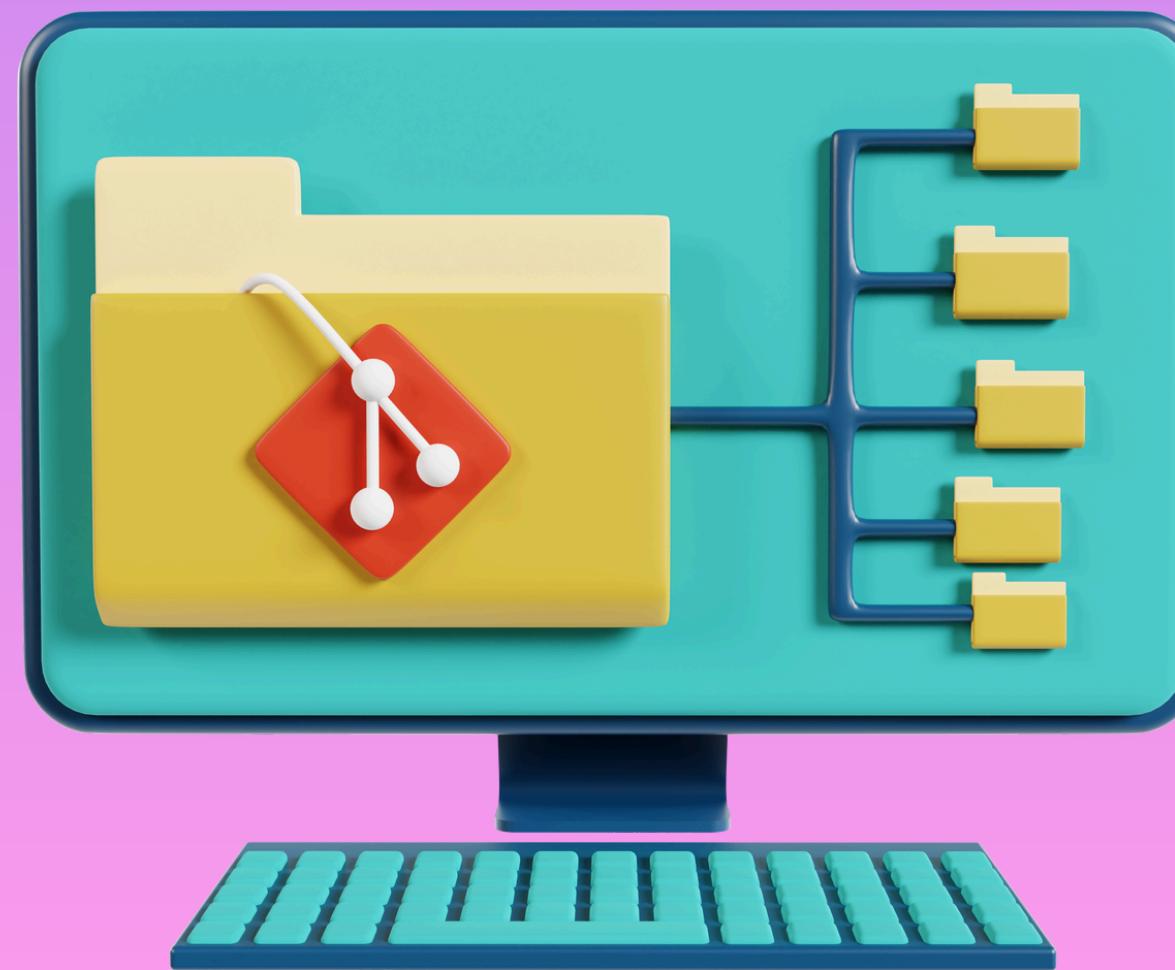


# VERSIONAMENTO



# Versionamento de código

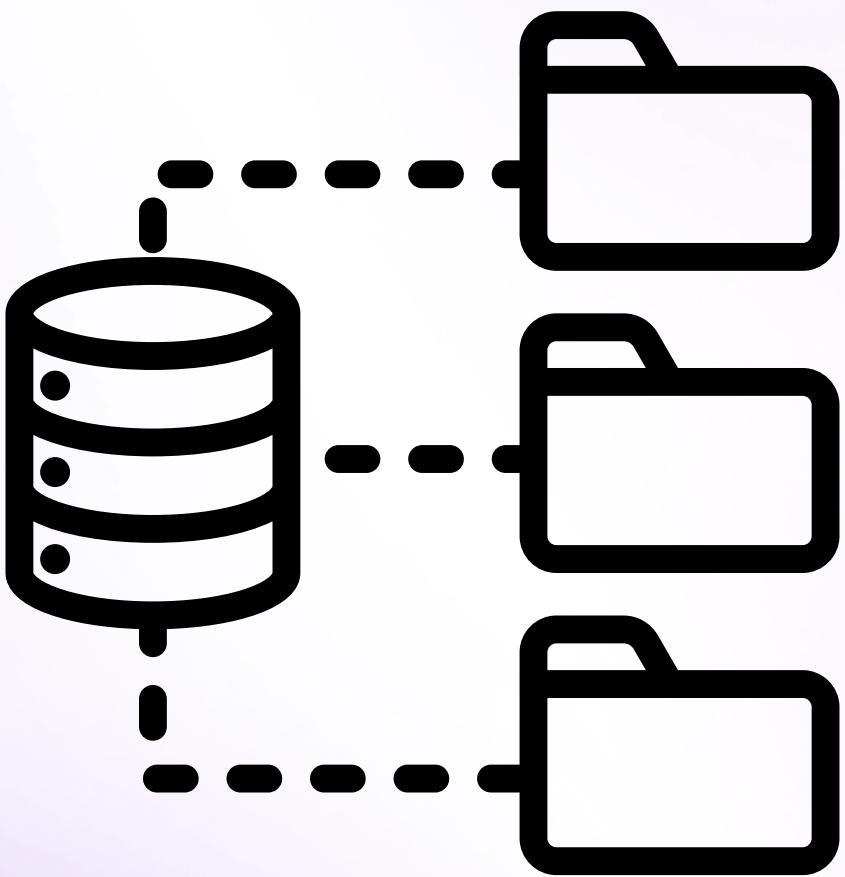
É uma maneira de **controlar** e **organizar** as diferentes versões de um código ou projeto de software ao longo do tempo.



# 3 vantagens do versionamento

- **Rastrear mudanças:** Cada vez que você faz uma alteração no código, ela é salva com um histórico detalhado (quem fez a alteração, o que foi alterado e quando).
- **Colaborar facilmente:** Se várias pessoas estiverem trabalhando no mesmo projeto, o versionamento ajuda a manter tudo organizado e evitar conflitos.
- **Reverter erros:** Se algo quebrar no código, você pode voltar para uma versão anterior que estava funcionando.

# REPOSITÓRIOS



# O que são repositórios?

De uma maneira simples, repositórios são os lugares onde você guarda os arquivos dos seus projetos :)



# Repositórios podem ser:

#Locais

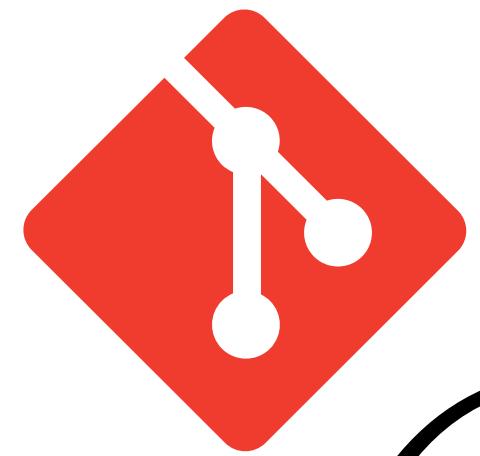


#Remotos



# GIT VS GITHUB

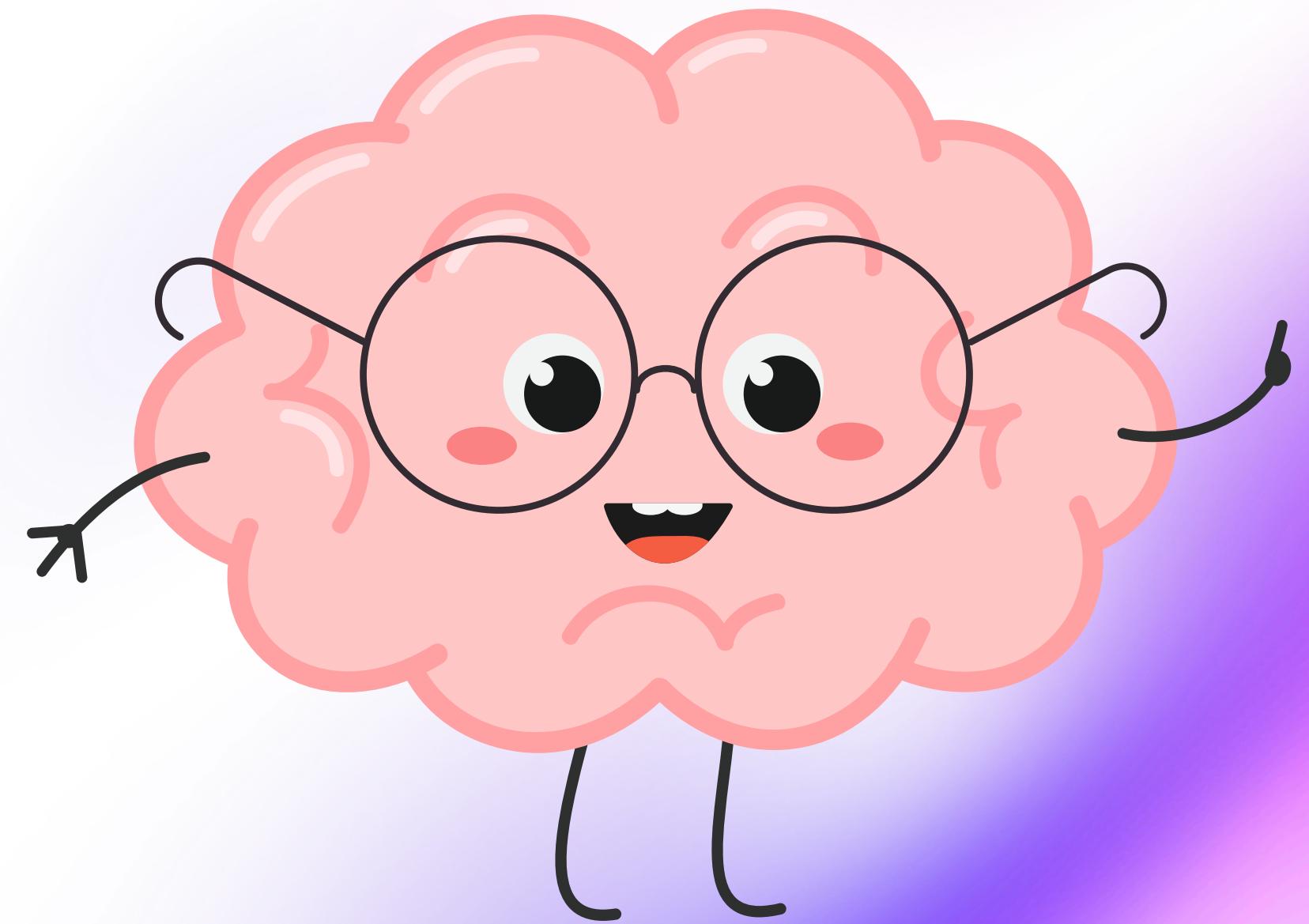


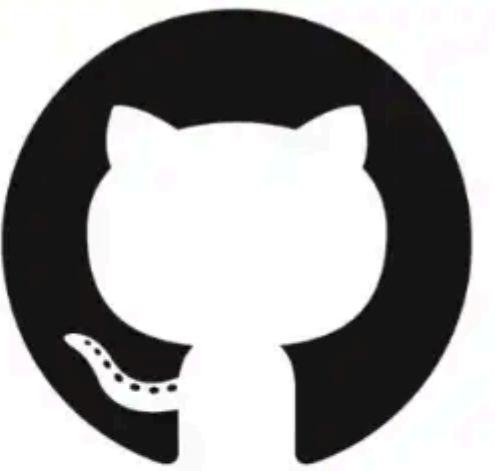


git

Ferramenta de  
versionamento

Repositórios Locais

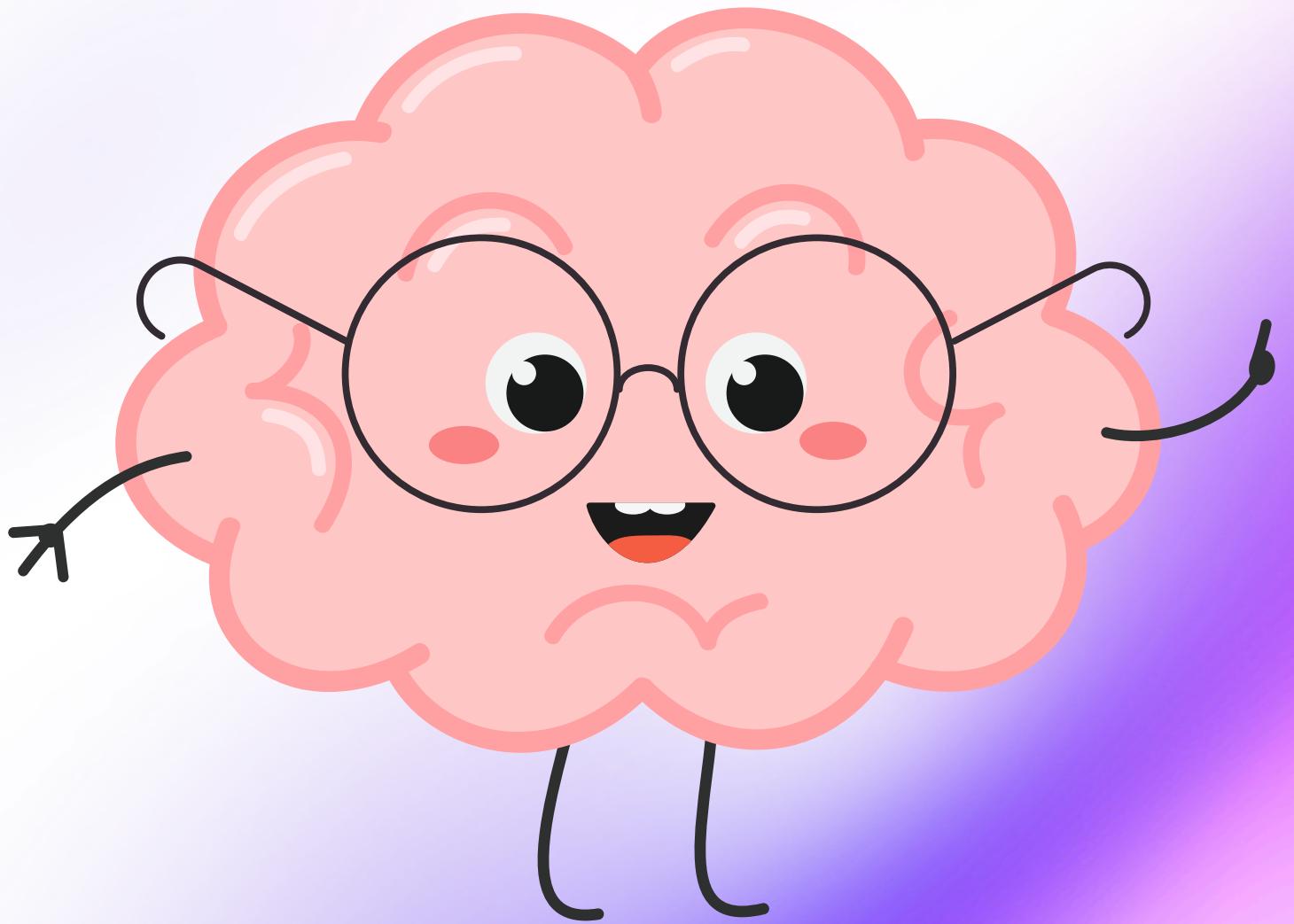




GitHub

Repositórios Remotos

Ferramenta de  
versionamento



# Resumo



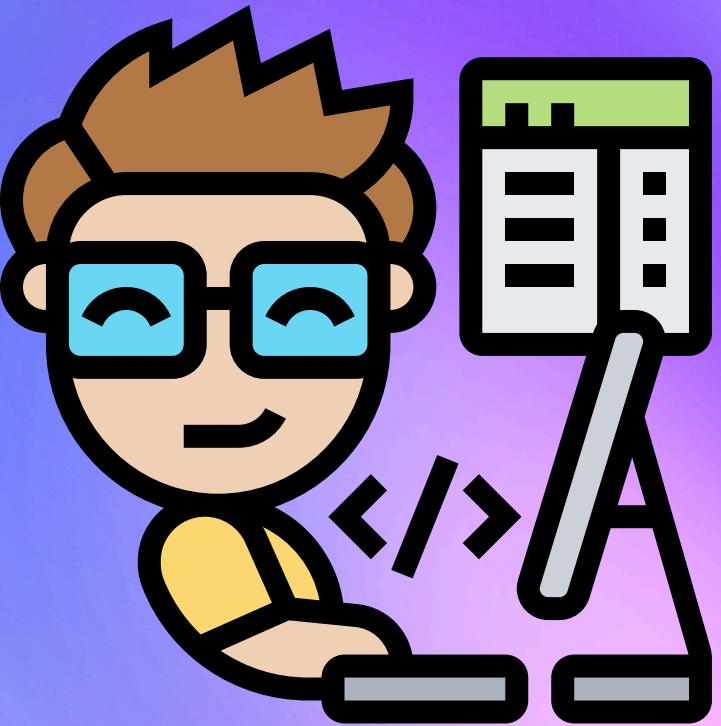
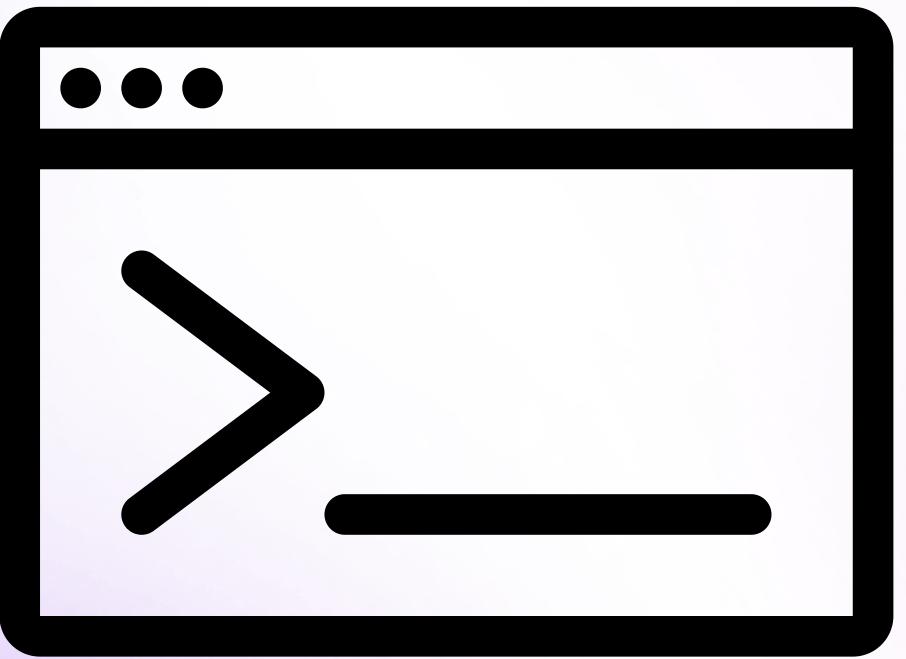
**Git**



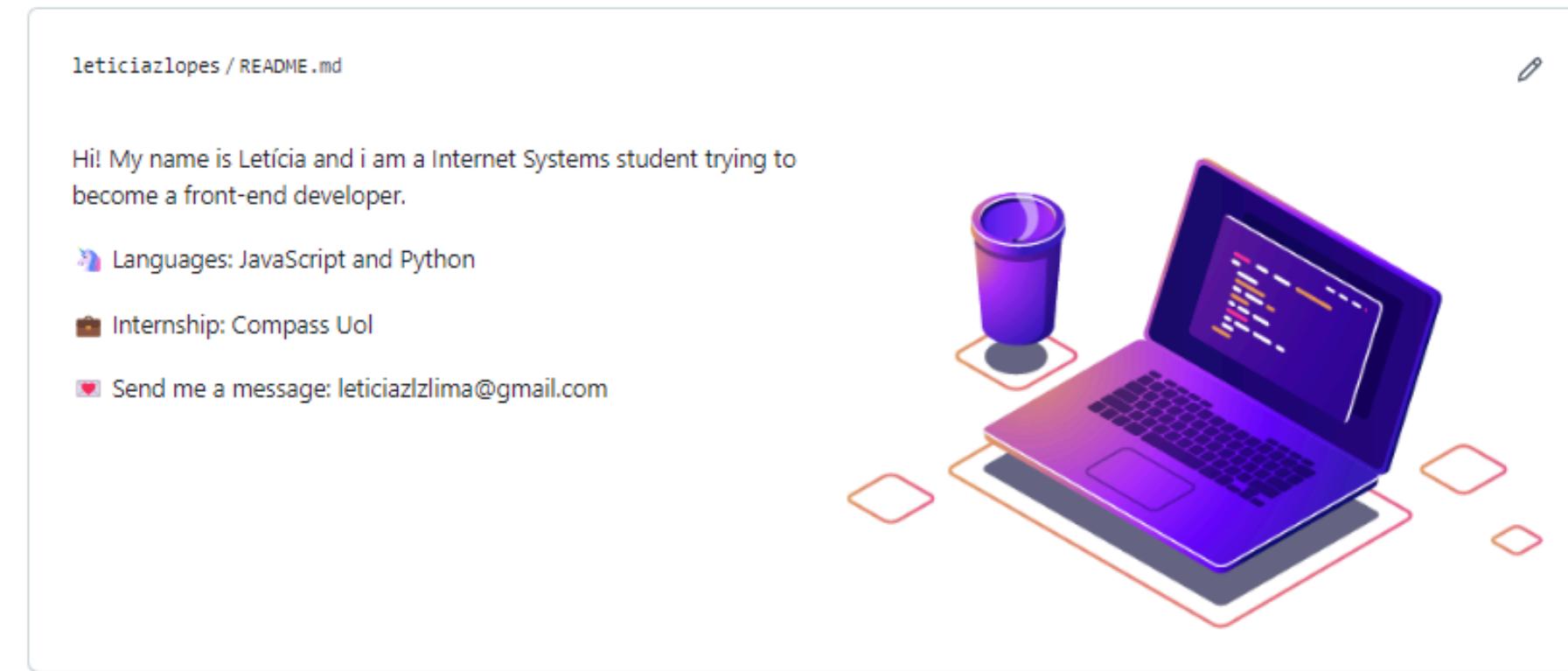
**GitHub**

| O QUE FAZ?   | ONDE RODA?                        |
|--|-----------------------------------|
| Guarda o histórico das mudanças no código, ajuda a organizar e colaborar no desenvolvimento. | No seu computador (localmente)    |
| Hospeda repositórios na nuvem, facilita o compartilhamento e a colaboração.                  | Na internet (repositório remoto). |

# AMBIENTE E COMANDOS



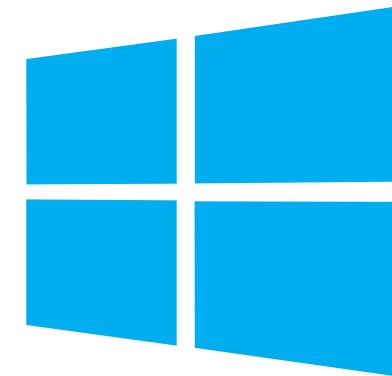
# Criando/Melhorando seu perfil!



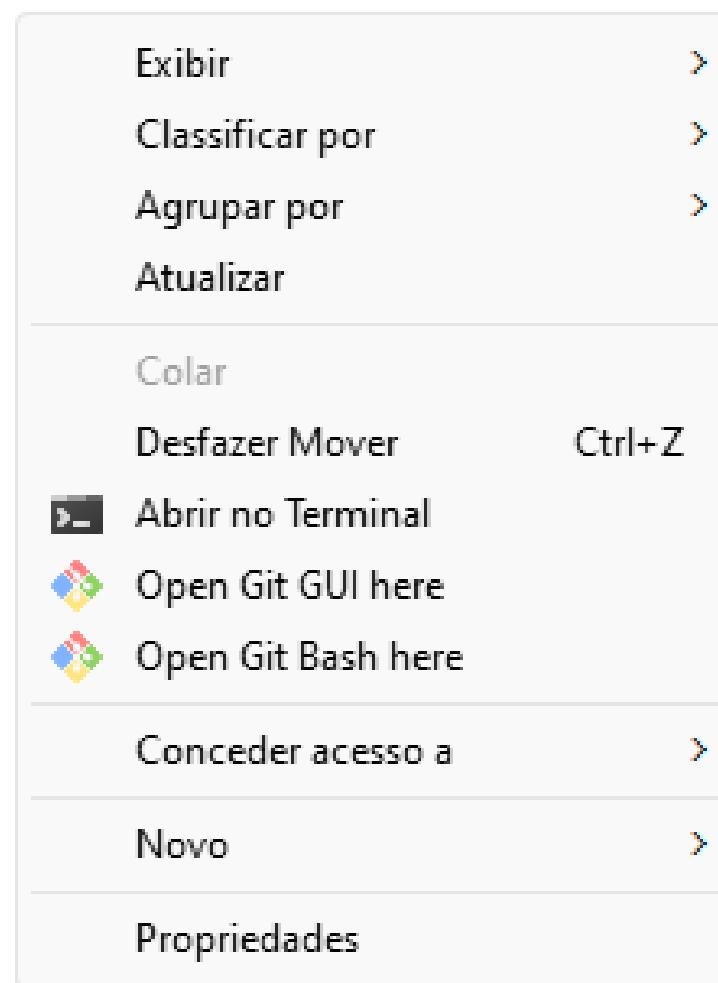
Se você ainda não tem um perfil, abra **[www.github.com](https://www.github.com)**:

1. Crie seu perfil;
2. Personalize com seus dados (**quanto mais informação, melhor**).

# Git bash



- O Git Bash é uma ferramenta que funciona como um "terminal" ou "linha de comando" Linux

A screenshot of a Git Bash terminal window titled "MINGW64:/c/Users/letic/OneDrive/Desktop". The window shows a command prompt: "Letic@LAPTOP-LETICIA MINGW64 ~/OneDrive/Desktop (main)" followed by a "\$" sign. The background of the terminal is black.

# Terminal



- Em sistemas operacionais baseados em Linux, você consegue utilizar o git pelo próprio terminal

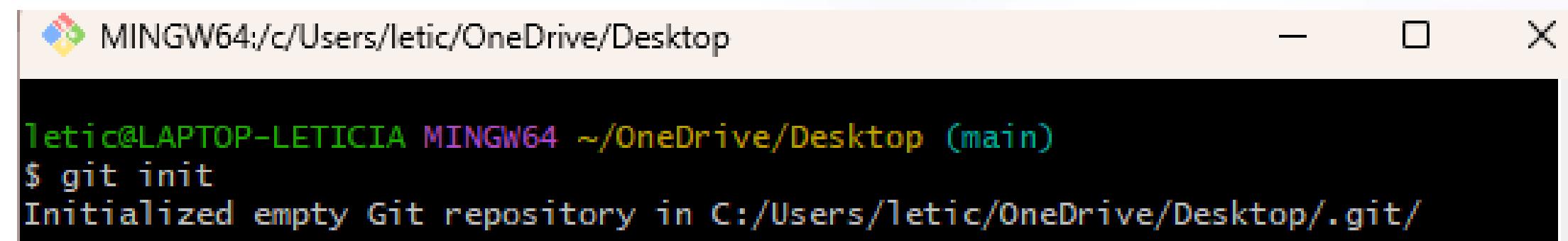
A screenshot of a terminal window titled 'suetone-carneiro@suetone: ~'. The window shows the command 'git --version' being run, with the output 'git version 2.40.1'. The terminal has a dark background and light-colored text. The window title bar includes standard icons for maximizing, minimizing, and closing the window.

```
suetone-carneiro@suetone:~$ git --version
git version 2.40.1
suetone-carneiro@suetone:~$ █
```

# Como criar um repositório?

- Na pasta escolhida, vá para seu git bash e digite:

git init



```
MINGW64:/c/Users/letic/OneDrive/Desktop
letic@LAPTOP-LETICIA MINGW64 ~/OneDrive/Desktop (main)
$ git init
Initialized empty Git repository in C:/Users/letic/OneDrive/Desktop/.git/
```

# Como checar se algo foi alterado?

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index.html
    modified:   styles.css

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    new_script.js

no changes added to commit (use "git add" and/or "git commit -a")
```

git status

# Como adicionar arquivos antes de fazer seu “checkpoint”(commit)

```
git add nome do arquivo.ext
```

Para adicionar todos de uma vez:

```
git add .
```

# O que é e como fazer um commit?

- É como "**tirar uma foto**" do estado atual de seu projeto
  - Essa foto **contém**:
    - Identificação única (hash)
    - Mensagem descritiva

```
git commit -m "sua descrição"
```



# A mensagem do **commit**

- Deve ser: **clara, concisa e descritiva.** Já pensou colocar como mensagem algo aleatório e não entender seu próprio commit?
- Quando colocados em repositório remotos, nossos commits servirão para outros developers entenderem nossas modificações!

# A mensagem do commit

- Use **verbos no imperativo**:
    - Remove, Corrige, Atualiza...
  - Commits **semânticos**:
- 
- `feat` - Commits do tipo `feat` indicam que seu trecho de código está incluindo um novo recurso (se relaciona com o MINOR do versionamento semântico).
  - `fix` - Commits do tipo `fix` indicam que seu trecho de código commitado está solucionando um problema (bug fix), (se relaciona com o PATCH do versionamento semântico).

| Comando Git   | Resultado no GitHub  |
|---|--|
| <code>git commit -m ":tada: Commit inicial"</code>                |  Commit inicial                 |
| <code>git commit -m ":books: docs: Atualização do README"</code>  |  docs: Atualização do README    |
| <code>git commit -m ":bug: fix: Loop infinito na linha 50"</code> |  fix: Loop infinito na linha 50 |

# README.md



- Principal fonte de documentação de um projeto;
- Fornece informações essenciais sobre o projeto, como uma descrição geral, instruções de instalação, uso, contribuições, e detalhes sobre as tecnologias utilizadas;

README.md

---

## Curriculum

Este projeto é um currículo simples desenvolvido em HTML e CSS. O objetivo é fornecer uma visualização clara e organizada das informações pessoais e profissionais do autor.

## Estrutura do Projeto

O currículo contém as seguintes seções:

- Nome: O nome do autor.
- Curso: O curso que o autor está cursando.
- Período: O período em que o autor está no curso.
- Cursos Adicionais: Inclui o curso de "Guia Básico de Git e GitHub".

# Como fazer um **clone** de um repositório?

```
git clone url-do-repositorio
```

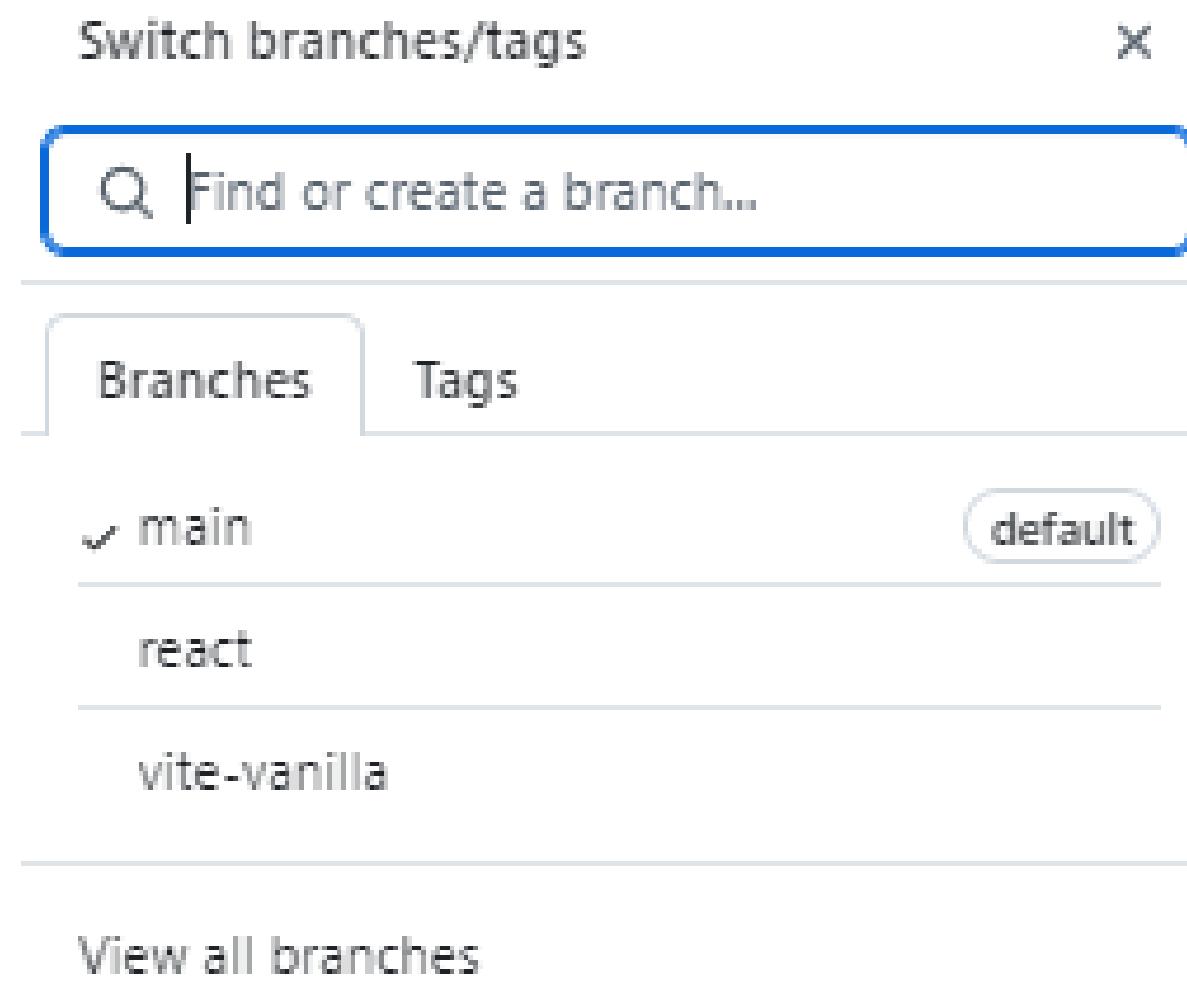
# Git pull e git push

- Pull: atualiza seu repositório das modificações; puxa
- Push: envia suas modificações ao repositório; empurra
- `git remote add origin https://github.com/<usuário>/<nome-do-repositorio>.git`

```
git push origin main
```

```
git pull origin main
```

# Git branch



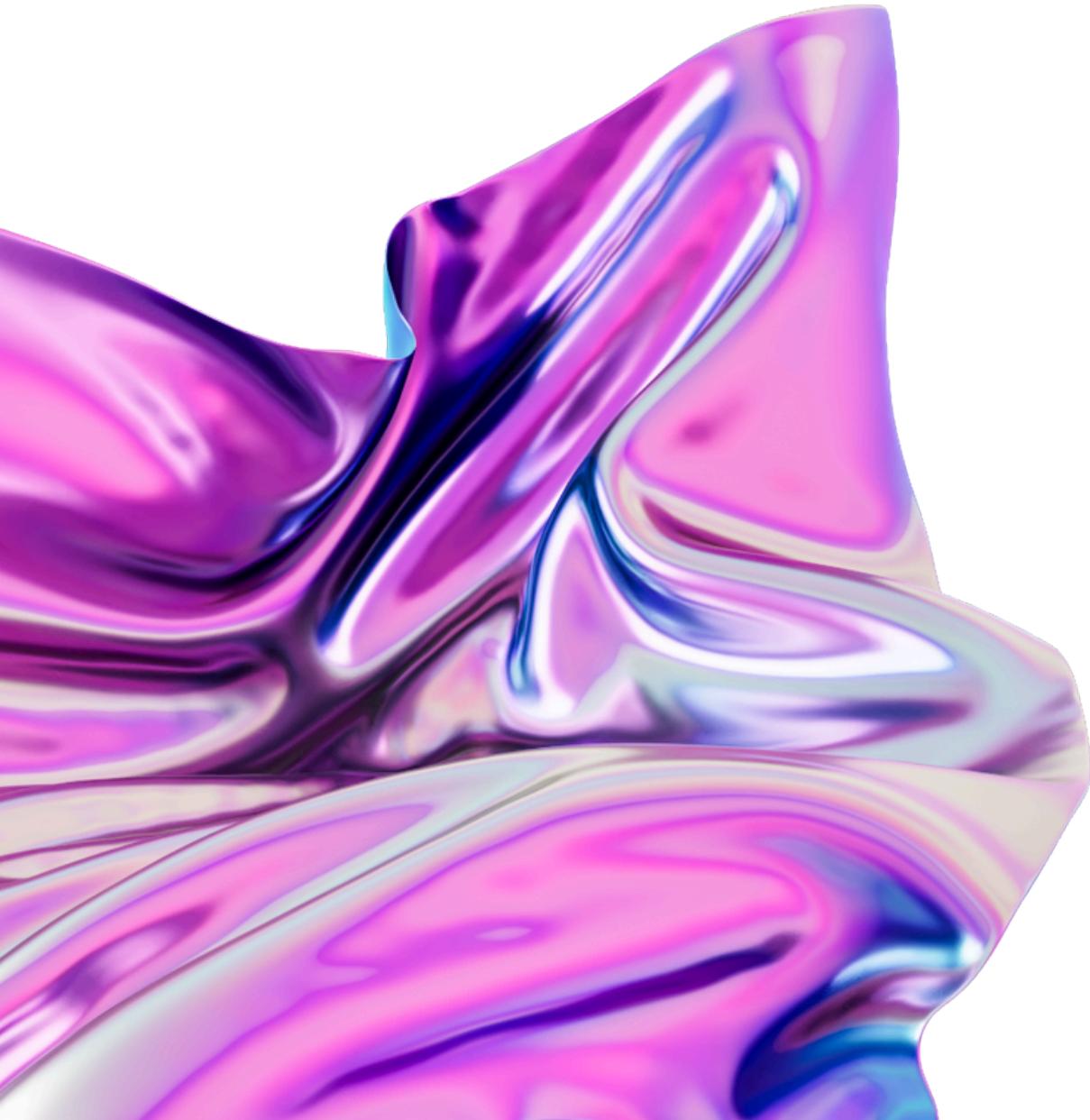
- Nos permite trabalhar com ramificações

`git branch -M main`

`git checkout nome`

`git checkout -b nome`

# Review



| COMANDO            | O <u>QUE FAZ?</u>   |
|--------------------|---|
| init               | Inicializa um repositório local   |
| status             | Mostra quais arquivos foram alterados                                   |
| add                | Adiciona os arquivos a uma área temporária                              |
| commit             | Registra as mudanças realizadas em relação ao estado anterior do código |
| remote add / clone | Traz os repositórios remotos para a sua máquina local.                  |

# PRÁTICA 1

**Seu Nome**

**Informações Pessoais**

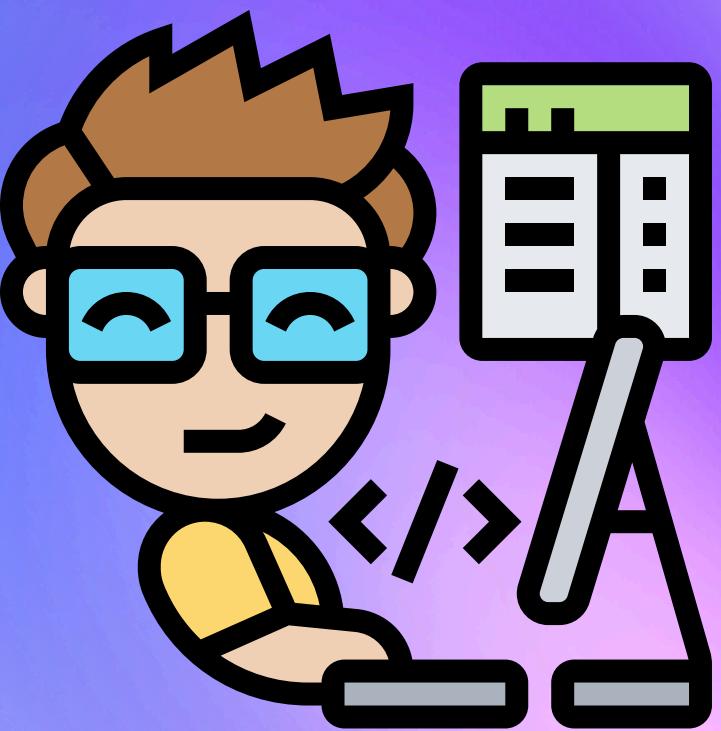
---

Curso: Tecnologia em Sistemas para Internet  
Período: adicione seu período

**Cursos**

---

- Guia de sobrevivência de Git e GitHub - Concluído



# CRIANDO SEU PRIMEIRO REPOSITÓRIO

- <https://github.com/new>

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*      Repository name \*

 leticiazlopes /

Great repository names are short and memorable. Need inspiration? How about [urban-succotash](#) ?

Description (optional)

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

 Private  
You choose who can see and commit to this repository.

Initialize this repository with:

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

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

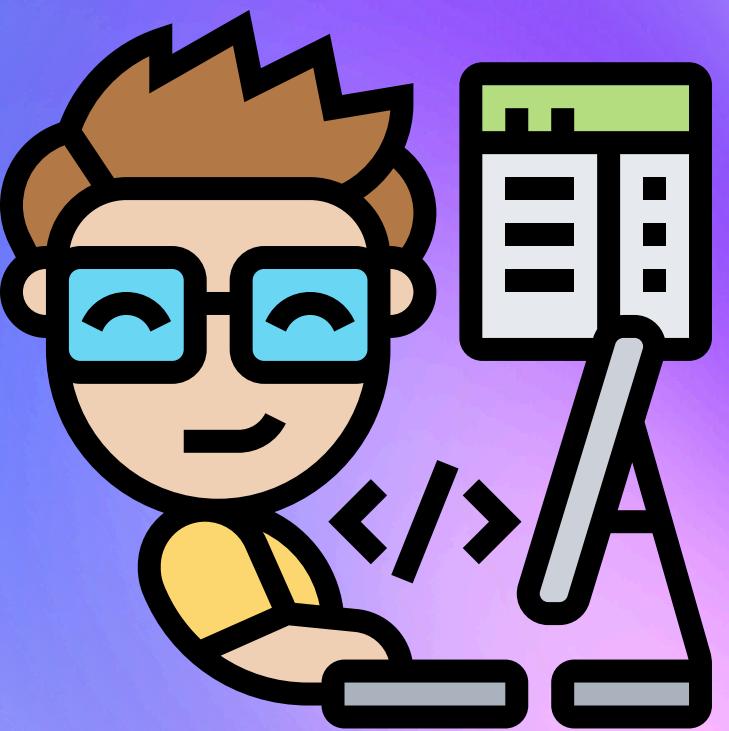
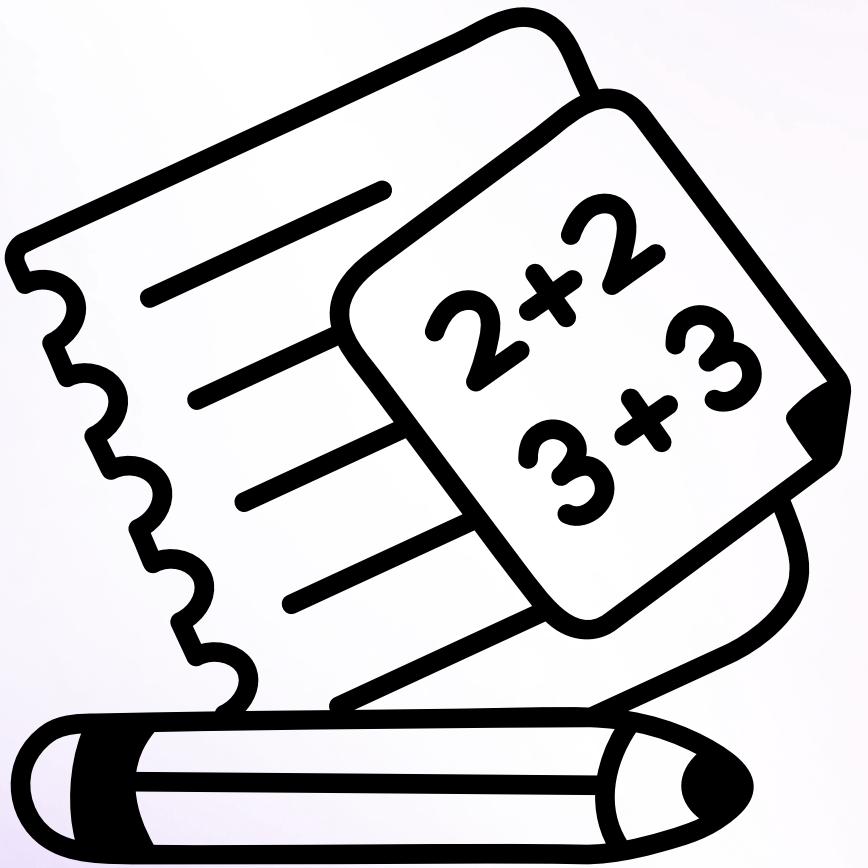
# CRIANDO SEU PRIMEIRO REPOSITÓRIO

- Em sua máquina, abra o git bash/terminal e use o comando **git init**;
- **git branch -M main**
- **git remote add origin https://github.com/<usuario>/<nome-do-repositorio>.git**
- **git clone https://github.com/leticiazlopes/guia-basico-git.git**
- Faça suas modificações no arquivo;

# CRIANDO SEU PRIMEIRO REPOSITÓRIO

- **git status**
- **git add .**
- **git commit -m ":sparkles: feat: Página de Currículo"**
- no vscode, personalize o arquivo "README.md" presente em `pratica01`, deixando externo em seus arquivos
- **git commit -m ":books: docs: Atualização do README"**
- **git push -u origin main**

# PRÁTICA 2



# IFPB PROJECTS

- <https://github.com/ifpb/projects>
- <https://github.com/ifpb/projects/tree/main/src/content/people>

Fork -> edição -> pull request

Quem já tem o perfil, vai ser monitor!

# BÔNUS

