



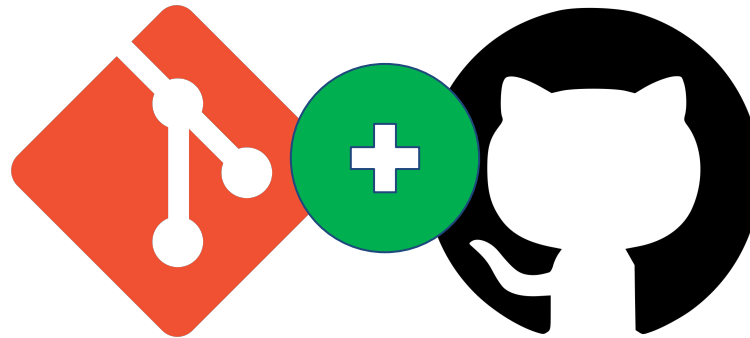
# Técnico em Informática



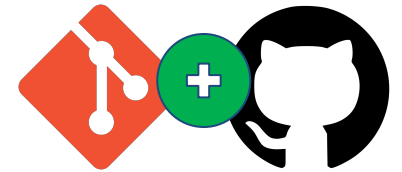
# Objetivo Geral

- **Inovação:** Estimular a criatividade e a aplicação de soluções tecnológicas inovadoras, preparando os alunos para os desafios do mercado de TI.
- **Gestão de Projetos:** Introduzir metodologias que facilitam o planejamento, execução e acompanhamento de projetos, promovendo a organização e eficiência.
- **Colaboração:** Capacitar os alunos a trabalhar de forma eficaz em equipe, utilizando ferramentas como Git e GitHub para controle de versão e gerenciamento de código.

# Git e GitHub

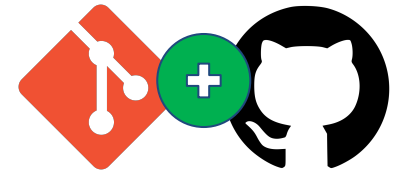


# Git e GitHub



**Para que  
serve isso?**

# Git ≠ GitHub



Plataforma de Rede Social  
para Programadores

Software de Controle de  
Versão (VCS)

# Git



# versionamento

# Git



## versões

# Git



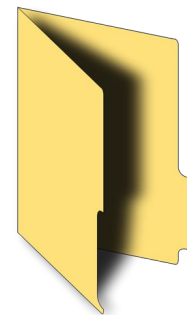
muitas  
**versões**  
de uma  
mesma  
coisa



# Exemplo



**index.html**



**site-cliente**

# Exemplo



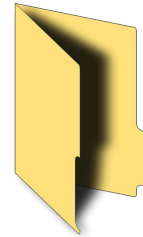
**site-cliente**



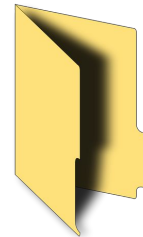
**site-cliente.zip**



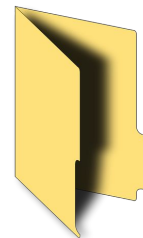
**site-clientev2.zip**



**site-cliente-proble  
ma.zip**

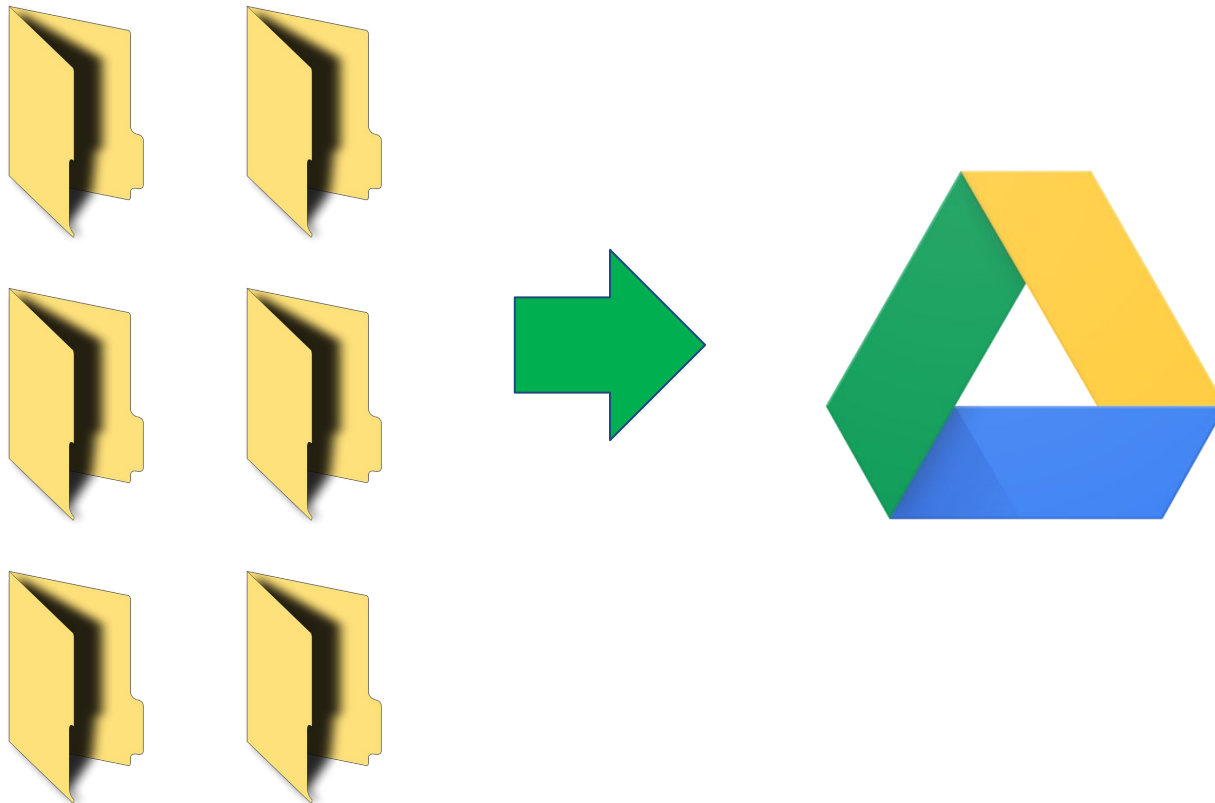


**site-cliente-agora-v  
ai.zip**

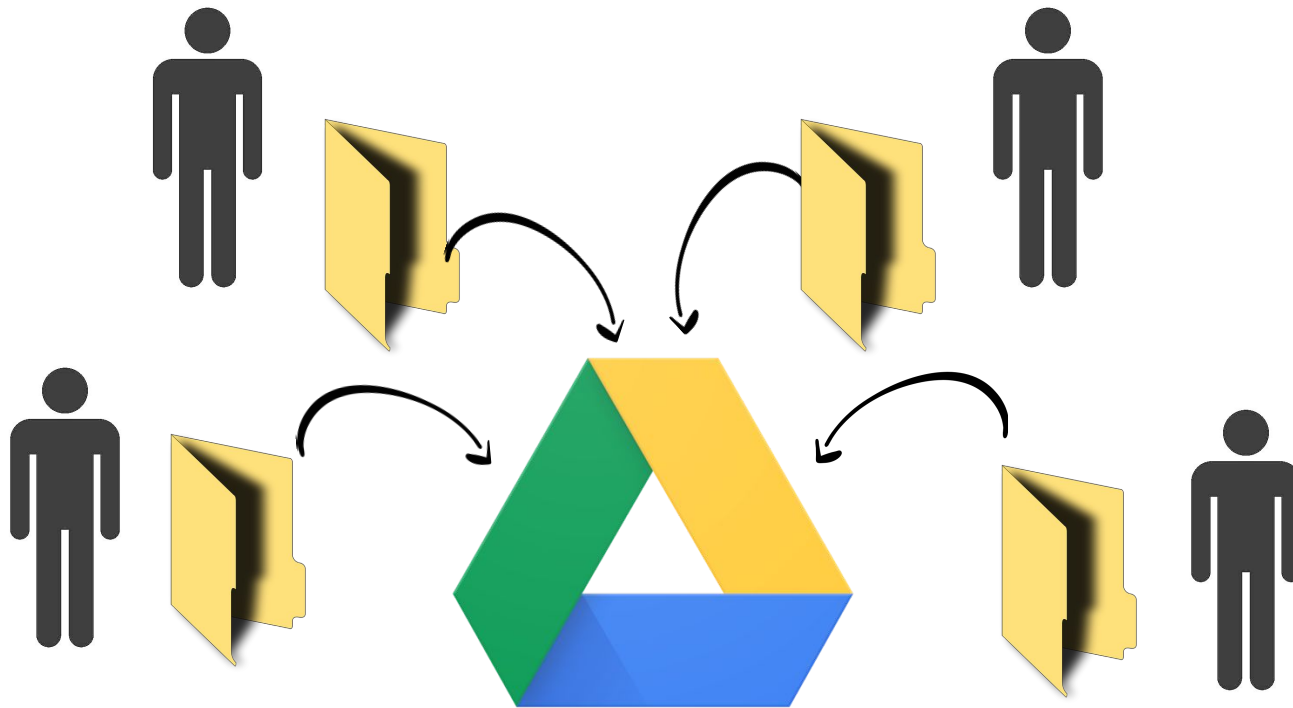


**site-cliente-mudo  
u-tudo.zip**

# Exemplo



# Exemplo



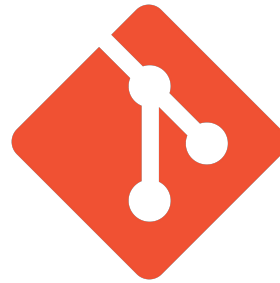
# Git



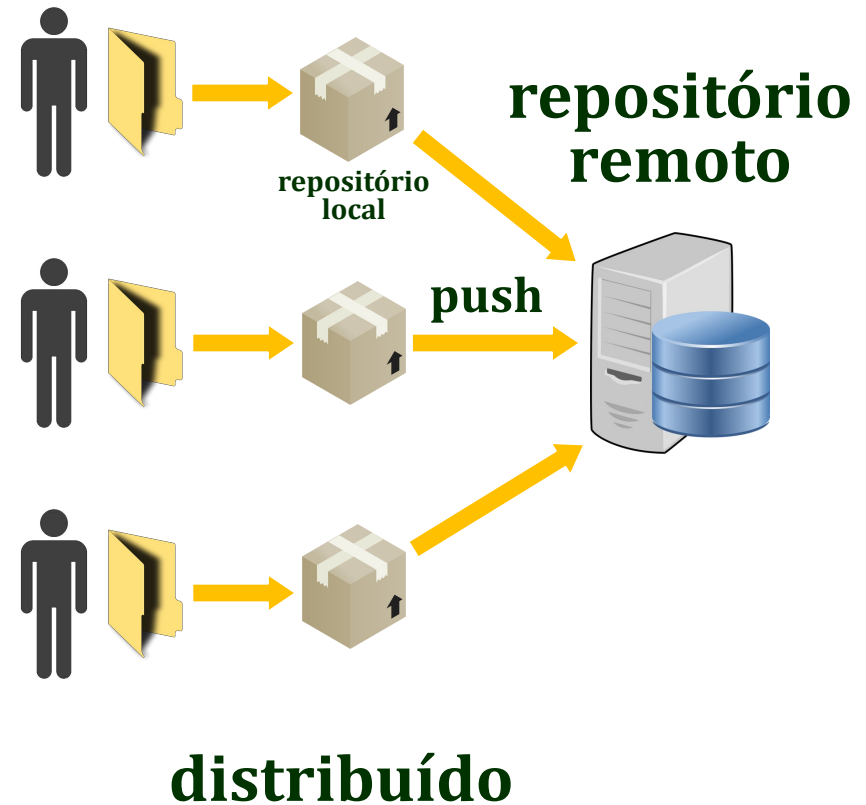
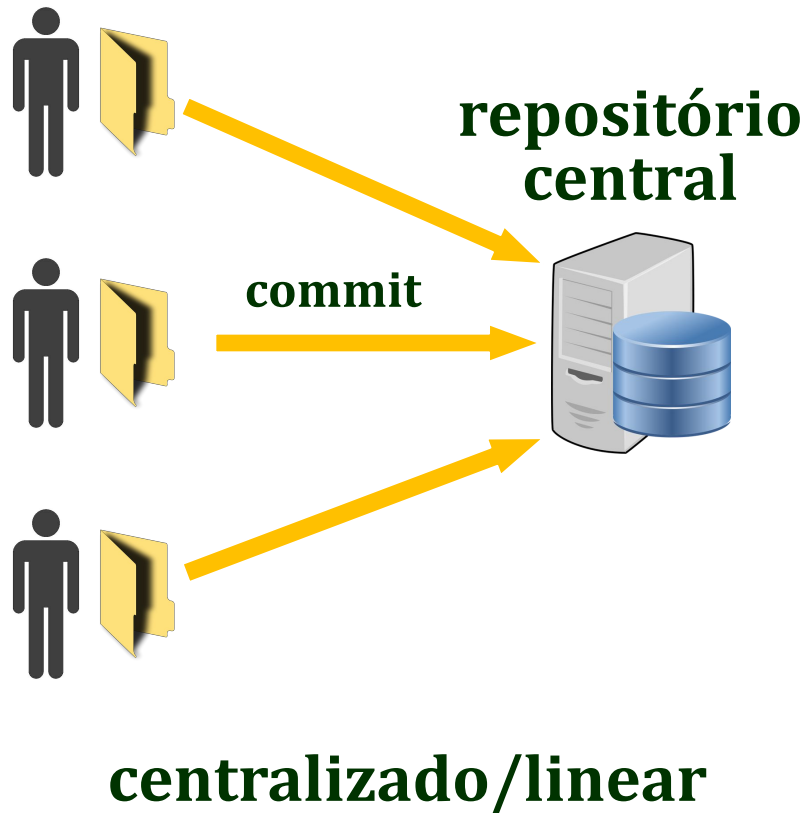
## Controle de Versão (VCS)

# Git

## versionamento



# Versionamento(VCS)

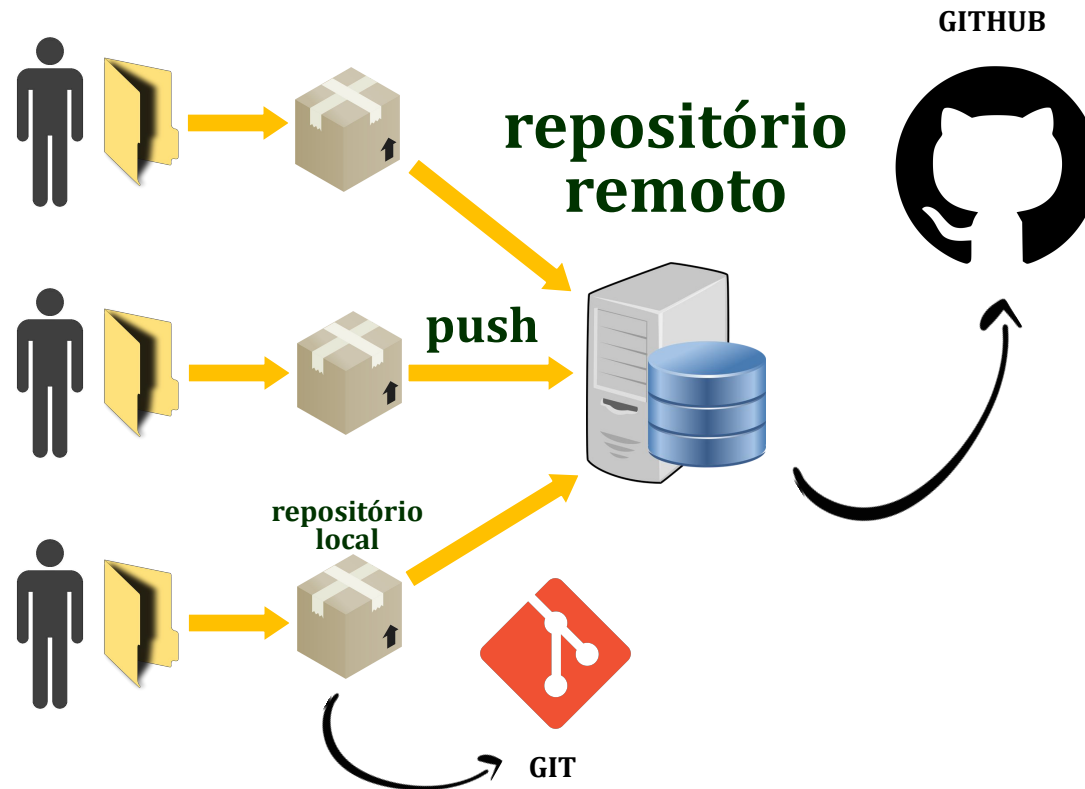


# Principais Vantagens

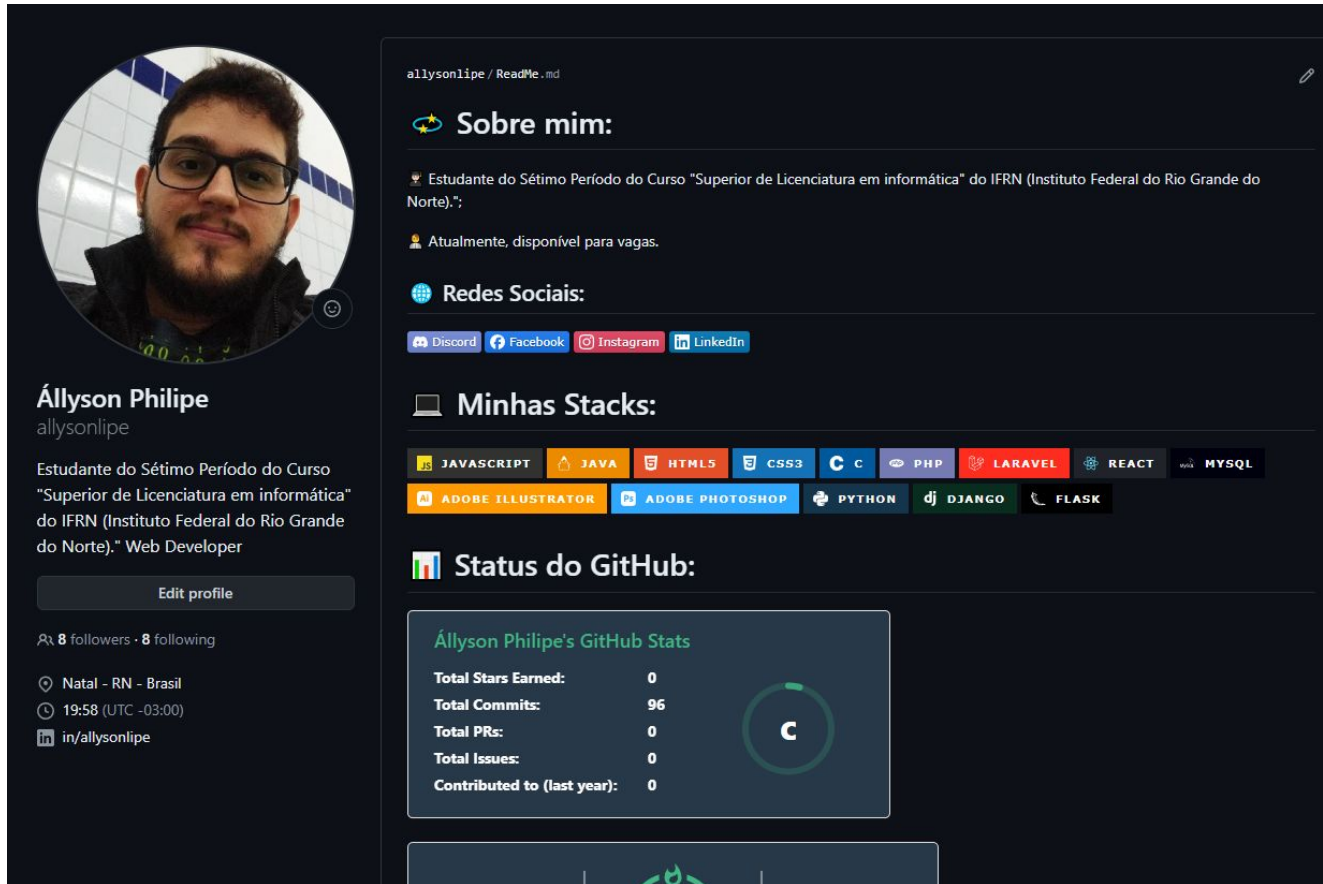
- Controle de histórico;
- Trabalho em equipe;
- Ramificação do Projeto;
- Segurança;
- Organização.



# Git ≠ GitHub



# GitHub.com/allysonlipe



The screenshot shows the GitHub profile of allysonlipe. The profile includes a circular profile picture of a man with glasses and a beard. The name 'Állyson Philipe' is displayed above the username 'allysonlipe'. The bio states: 'Estudante do Sétimo Período do Curso "Superior de Licenciatura em informática" do IFRN (Instituto Federal do Rio Grande do Norte)." Web Developer'. Below the bio is an 'Edit profile' button. The profile also shows '8 followers · 8 following'. Location is 'Natal - RN - Brasil', time zone is '19:58 (UTC -03:00)', and the GitHub handle is 'in/allysonlipe'. The 'Sobre mim:' section mentions being a student at IFRN and being available for jobs. The 'Redes Sociais:' section lists links for Discord, Facebook, Instagram, and LinkedIn. The 'Minhas Stacks:' section displays a row of technology icons including JavaScript, Java, HTML5, CSS3, C, PHP, Laravel, React, MySQL, Adobe Illustrator, Adobe Photoshop, Python, Django, and Flask. The 'Status do GitHub:' section shows 'Állyson Philipe's GitHub Stats' with a table of metrics and a circular commit counter.

Állyson Philipe's GitHub Stats	
Total Stars Earned:	0
Total Commits:	96
Total PRs:	0
Total Issues:	0
Contributed to (last year):	0

# O que é Oferecido

- Repositórios ilimitados;
- Hospedagem de código-fonte;
- Características de rede social;
- GitHub Pages integrado;
- Colaboração ;
- Forks.



# Técnico em Informática

# Vamos recapitular...

## versão

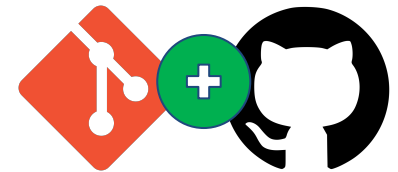
É uma instância específica de um conjunto de arquivos ou software em um determinado momento, refletindo o estado e as alterações feitas até aquele ponto. Cada versão captura uma "fotografia" do projeto, permitindo que se retorne a ela ou se compare com outras versões anteriores ou posteriores.

# Controle de Versões

# versionamento

Versionamento é o processo de gerenciar as diferentes versões de um conjunto de arquivos, especialmente em desenvolvimento de software. Ele permite registrar mudanças, acompanhar a evolução de um projeto ao longo do tempo e colaborar com outras pessoas sem perder o histórico das modificações.

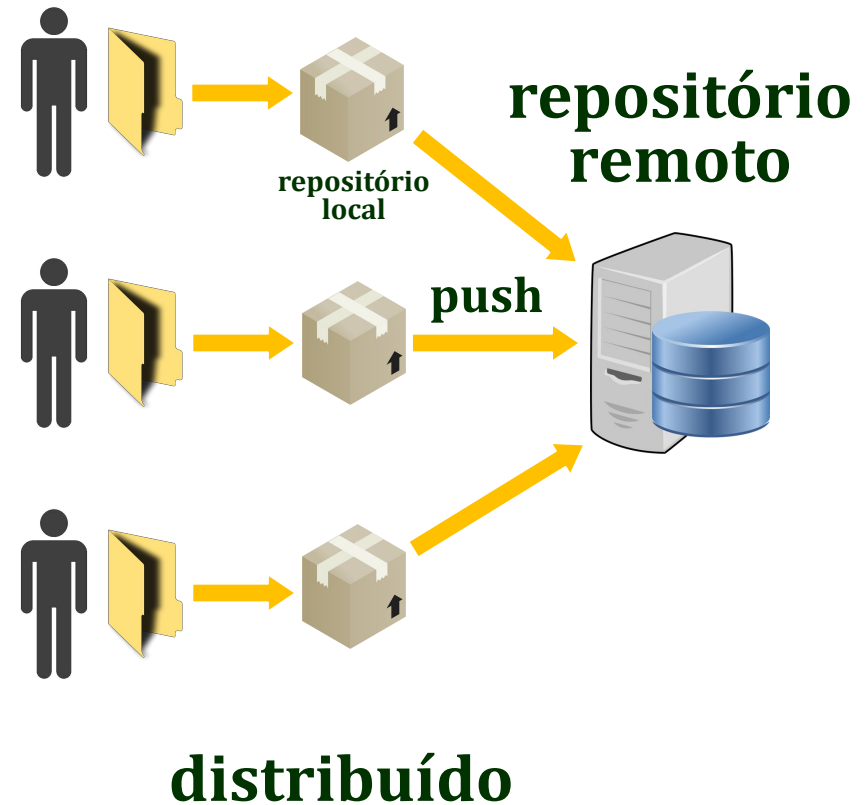
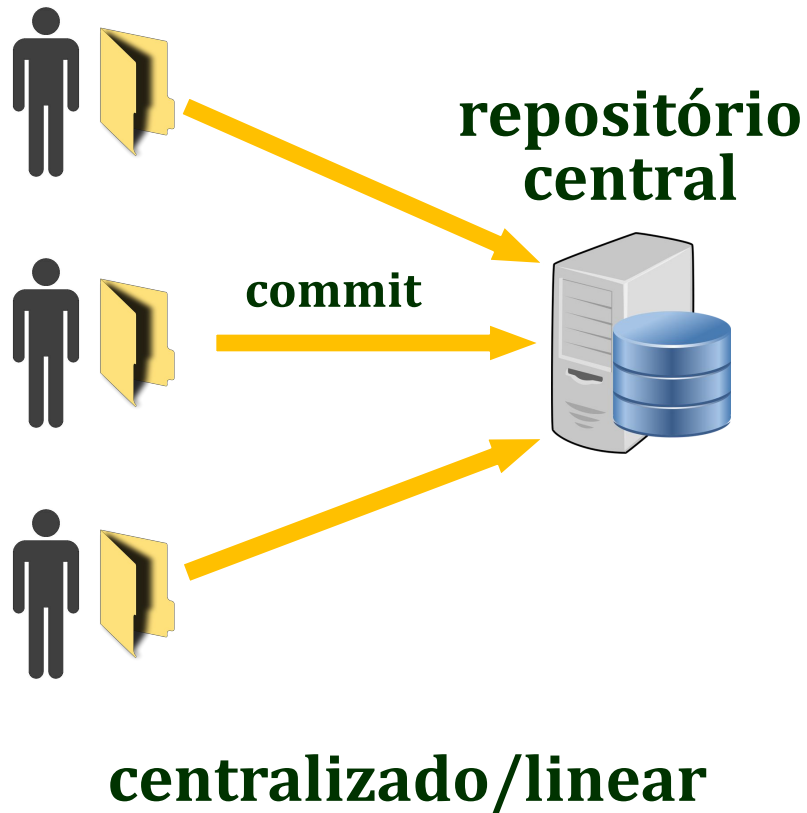
# Git ≠ GitHub



“Plataforma de Rede Social  
para Programadores”

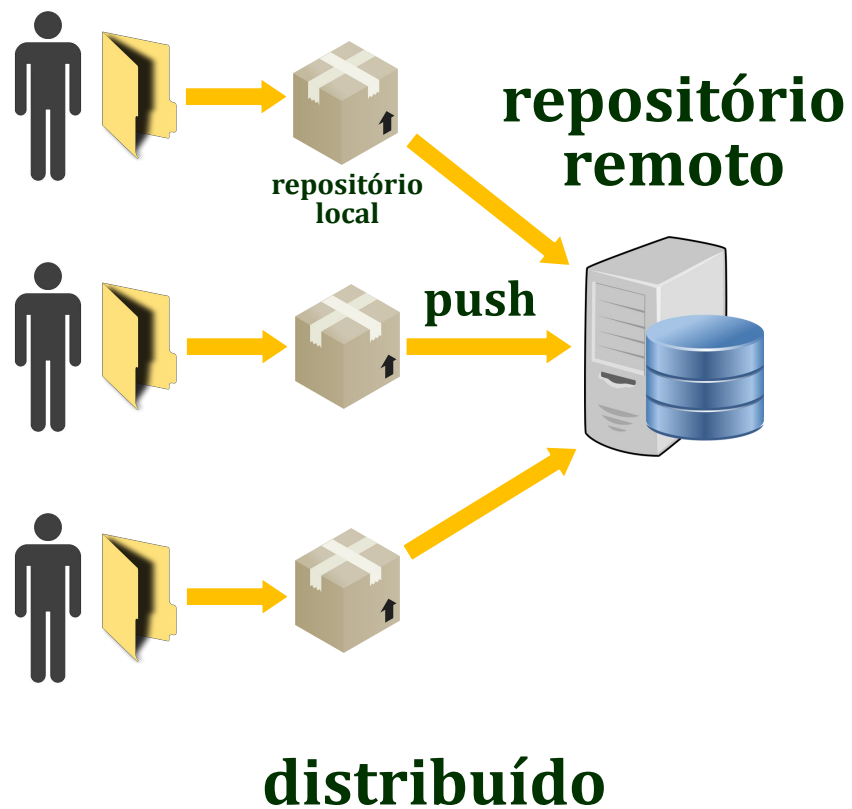
Software de Controle de  
Versão (VCS)

# Versionamento(VCS)

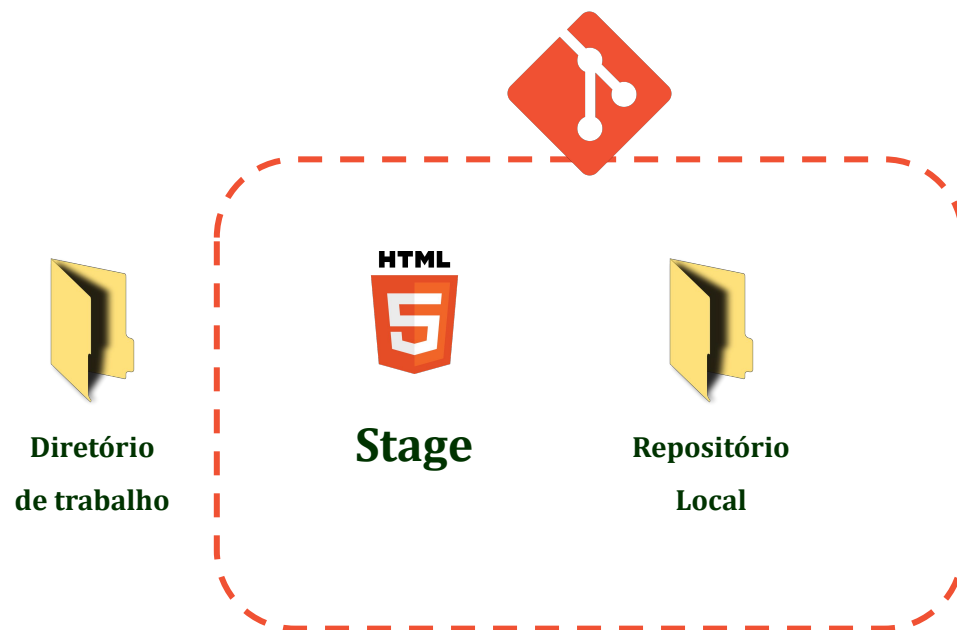




# git



# git



# Comandos git



- `git init`
- `git status`
- `git add <arquivo>` ou `git add .`
- `git rm --cached <arquivo>`
- `git commit -m "nome do commit"`
- `git log`
- `git push`
- `git restore`

# commit



Um ***commit*** no **Git** é um registro de mudanças feitas no código-fonte de um projeto. Ele funciona como um "***snapshot***" ou uma captura instantânea do estado do repositório em um determinado momento. Cada ***commit*** armazena as alterações realizadas nos arquivos, junto com uma mensagem descritiva, que explica o que foi modificado, adicionado ou removido.

- **git commit --amend -m "nome do commit"**

# Comandos git



## git restore - Restaura os arquivos

```
011.py x
resolucoes > 011.py > ...
1  # Faça um Programa que peça 2 números inteiros e um número real.
2  # Calcule e mostre:
3  #     o produto do dobro do primeiro com metade do segundo.
4  #     a soma do triplo do primeiro com o terceiro.
5  #     o terceiro elevado ao cubo.
6
7  n1 = int(input('Digite o primeiro número: '))
8  n2 = int(input('Digite o segundo número: '))
9  n3 = float(input('Digite o terceiro número: '))
10
11  calc1 = 2*n1*n2/2
12  print('O produto do dobro do primeiro com metade do segundo: ' + str(calc1))
13  calc2 = (3*n1)+n3
14  print('a soma do triplo do primeiro com o terceiro: ' + str(calc2))
15  calc3 = n3**3
16  print('o terceiro elevado ao cubo: ' + str(calc3))
```

# Comandos git



## git restore - Restaura os arquivos rastreados

```
011.py  X
resolucoes > 011.py > ...
1  # Faça um Programa que peça 2 números inteiros e um número real.
2  # Calcule e mostre:
3  #     o produto do dobro do primeiro com metade do segundo.
4  #     a soma do triplo do primeiro com o terceiro.
5  #     o terceiro elevado ao cubo.
6
7  n1 = int(input('Digite o primeiro número: '))
8  n2 = int(input('Digite o segundo número: '))
9  n3 = float(input('Digite o terceiro número: '))
10
11  calc1 = 2*n1*n2/2
12  print('O produto do dobro do primeiro com metade do segundo: '+ str(calc1))
13  calc2 = (3*n1)+n3
14  print('a soma do triplo do primeiro com o terceiro: '+ str(calc2))
15  calc3 = n3**3
16  print('o terceiro elevado ao cubo: '+str(calc3))
17
18  #esse texto será revertido.
```

# Comandos git



## git restore - Restaura os arquivos rastreados

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
• PS C:\Users\ALLYSON\Desktop\estudos\estudosPython> git restore resolucoes/011.py  
○ PS C:\Users\ALLYSON\Desktop\estudos\estudosPython> 
```

```
14 print('a soma do triplo do primeiro com o terceiro: '+ str(calc2))  
15 calc3 = n3**3  
16 print('o terceiro elevado ao cubo: '+str(calc3))
```

# Comandos git



**git clean -** Restaura os arquivos não rastreados

**git clean -n:** Mostra os arquivos não rastreados que podem ser deletados.

**git clean -f:** Deleta os arquivos não rastreados.

**git clean -fd:** Deleta os arquivos rastreados e não **rastreados**.



# Comandos git



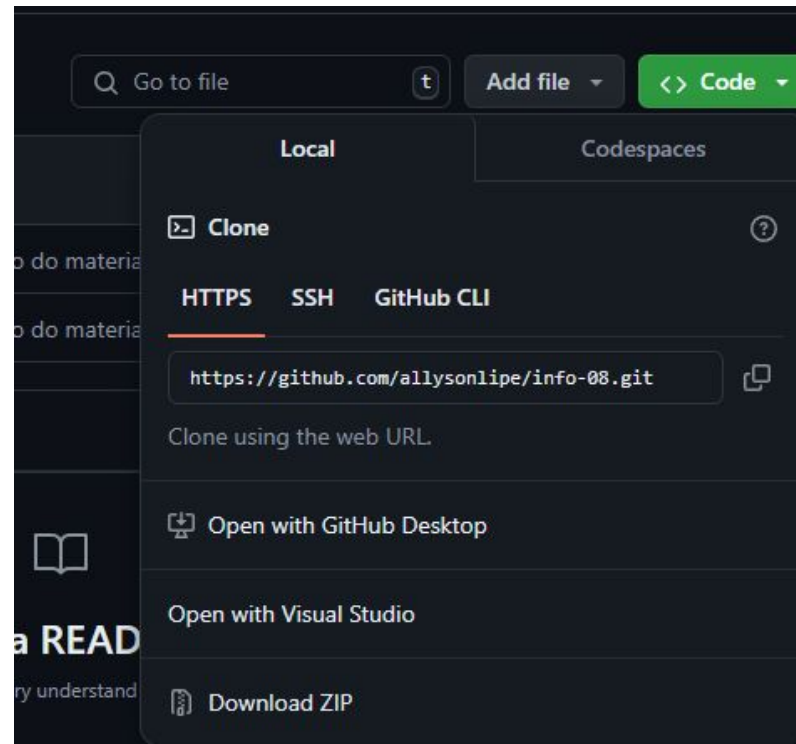
## Configuração

- `git config --list`
- `git config --global user.name "Seu Nome"`
- `git config --global user.email "seuemail@exemplo.com"`

# Comandos git



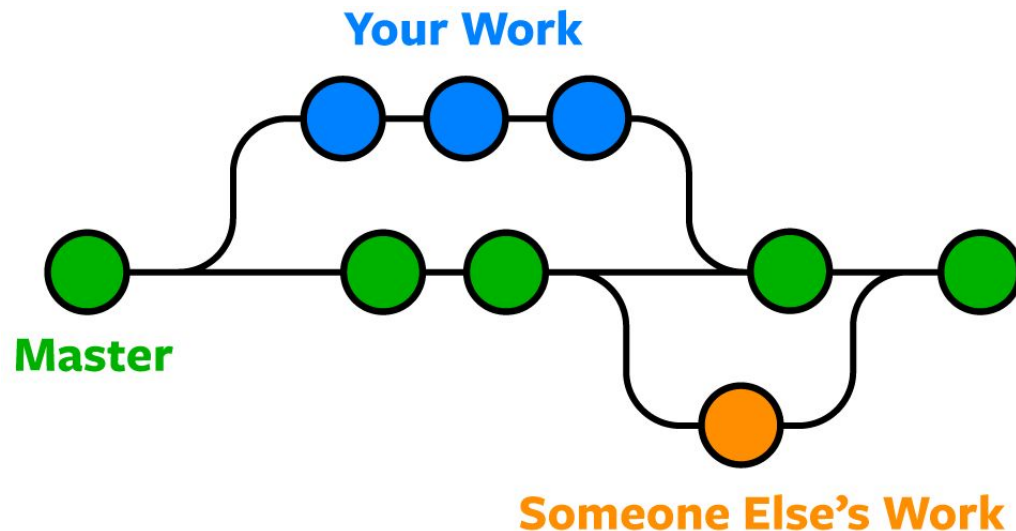
- **git clone:** Clona um repositório em um novo diretório  
**`https://github.com/allysonlipe/info-08.git`**



# BRANCH



Uma **branch** (ou "ramificação") é um conceito fundamental no **Git**, que representa uma linha independente de desenvolvimento dentro de um repositório. Cada branch permite que você trabalhe em diferentes **versões** do seu projeto simultaneamente, sem afetar a versão principal (geralmente chamada de **main** ou **master**).



# BRANCH

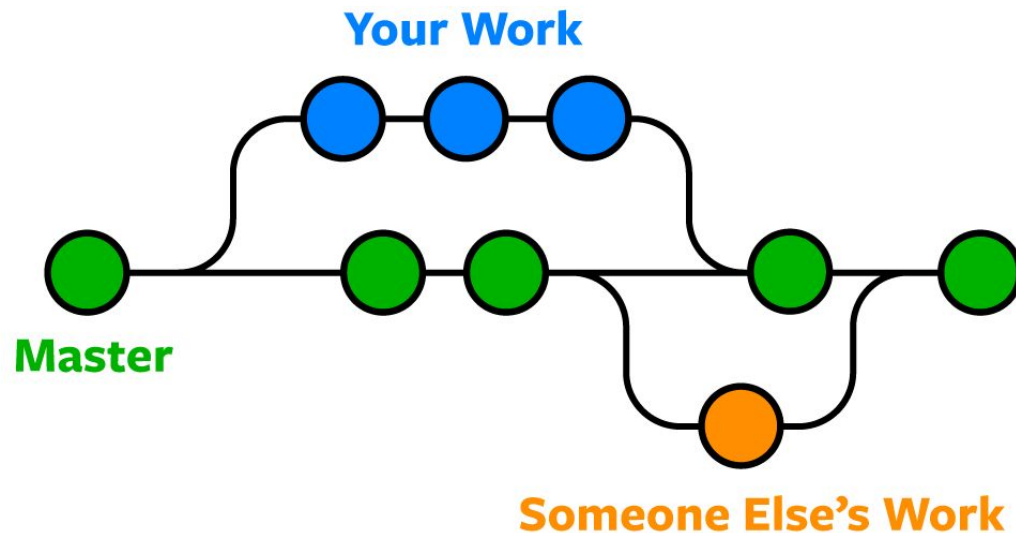


1. **Linha de Desenvolvimento Independente:** Cada **branch** é uma linha de desenvolvimento separada, o que permite que você trabalhe em novas funcionalidades, correções de bugs, ou experimentos sem interferir na versão estável do projeto.
2. **Ponto de Partida e Mergulho:** As branches geralmente começam como uma cópia de outra branch (por exemplo, main) e podem divergir a partir daí. As alterações feitas em uma branch podem, posteriormente, ser fundidas (ou "merged") de volta à branch original ou a outras branches.
3. **Trabalho em Equipe:** Em equipes de desenvolvimento, diferentes membros podem trabalhar em diferentes branches ao mesmo tempo, facilitando a colaboração sem causar conflitos no código. Por exemplo, um desenvolvedor pode estar trabalhando em uma branch para uma nova funcionalidade, enquanto outro corrige um bug em outra branch.
4. **Git e Versionamento:** O Git mantém um histórico de todos os commits em cada branch, permitindo que você veja exatamente quais mudanças foram feitas, por quem e quando.

# BRANCH



**Branch Principal** (*main* ou *master*): A branch onde geralmente reside o código de produção ou a versão estável do projeto.



# Gerenciamento de Branches:



- Criar uma nova branch:

**git branch <nome-da-branch>**

**git checkout -b <nome-da-branch>**

# Gerenciamento de Branches:



- Deletar uma Branch:

**git branch -D <nome-da-branch>**

# Gerenciamento de Branches:



- **Mudar para Outra Branch:**

**git checkout <nome-da-branch>**



# Gerenciamento de Branches:



- **Fundir (Merge) uma Branch:**

**git checkout main**

**git merge nova-branch**

# Bônus:



## **.gitignore e README**

### **Por que usar o .gitignore?**

1. **Manter o Repositório Limpo:** Evita que arquivos temporários ou específicos da máquina, como arquivos de configuração de IDEs, caches, logs ou binários, sejam versionados.
2. **Proteção de Informações Sensíveis:** Impede que informações sensíveis, como senhas, chaves de **API**, ou arquivos de configuração locais, sejam incluídas no repositório.
3. **Facilidade de Colaboração:** Evita conflitos de arquivos que não deveriam ser compartilhados entre diferentes desenvolvedores.

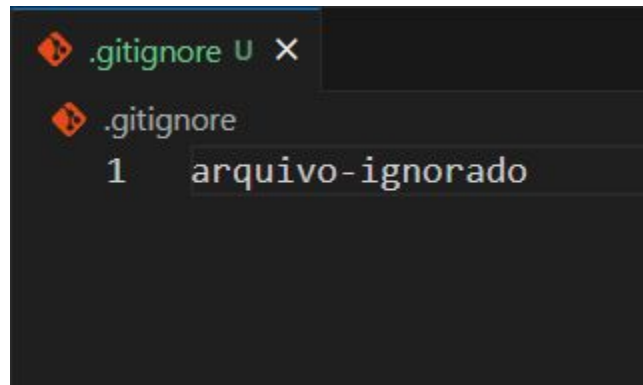


# Bônus:

## **.gitignore e README**

### **Como Funciona?**

O arquivo `.gitignore` funciona com base em padrões de nomeação e caminhos. Você pode listar arquivos, diretórios ou padrões globais para dizer ao Git o que deve ser ignorado.

A screenshot of a code editor with a dark background. The top tab is labeled '.gitignore U X'. The main editor area shows the text '.gitignore' on the first line and '1 arquivo-ignorado' on the second line. The '1' is highlighted with a mouse cursor.

```
.gitignore
1  arquivo-ignorado
```

# Bônus:



## .gitignore e README

### O que é README

O **README** é um arquivo fundamental em qualquer repositório de código. Ele fornece informações essenciais sobre o projeto, ajudando os usuários e colaboradores a entenderem o propósito, como usá-lo, e como contribuir para ele. Normalmente, o arquivo **README** é escrito em *Markdown* (com extensão .md), o que permite a formatação de texto de forma simples e organizada.