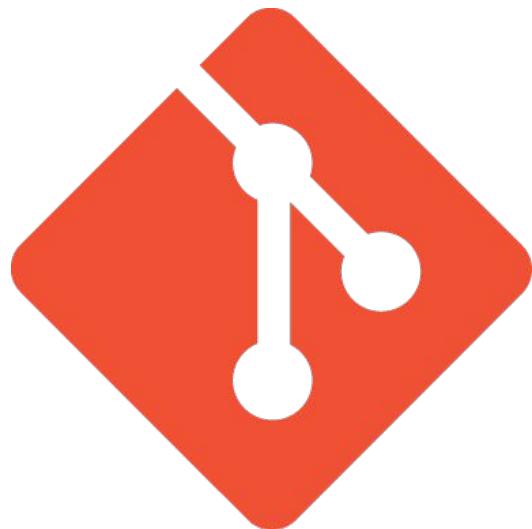
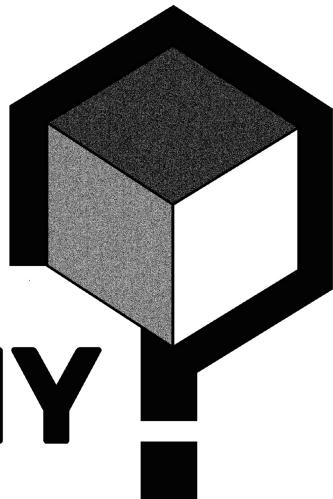


# Git e GitHub

Uma introdução ao versionamento de código e trabalho colaborativo

TÉO  
ME WHY?



27/01/2025

Git e GitHub © 2025 by Téo Calvo is licensed under CC BY-NC-SA 4.0.  
To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>



# Quem

## Teodoro “Téo” Calvo

- Bacharel em Estatística - FCT UNESP

- Especialista em Big Data & Data Science - UFPR

- Streamer - Twitch

- Desenvolvedor Backend - Globo

- Membro e Instrutor - Instituto Aaron Swartz



PARTNER



# Trajetória

2016 2017 2018 2019 2020 2021 2022 2023 2024 2025



sas®



Grupo  
Boticário



2019



2020



2021



2022



2023



2024



2025



# Agenda (1/2)

## 1. Introdução ao Git e Controle de Versão

- Explicação sobre a necessidade de controle de versão;
- Visão geral do Git e seus benefícios;
- Instalação do Git no ambiente Windows;

## 2. Setup

- Download
- Instalação
- Configuração Inicial

## 3. Introdução ao terminal

- O que é
- Comandos básicos no terminal

## 4. Iniciando um repositório

- Criação do diretório (projeto)
- O que há no diretório **.git** ?





# Agenda (2/2)

## 5. Ciclo mudanças

## 6. Trabalhando com Branches

- O que são branches?
- Criando, listando e excluindo branches;

## 7. Trabalhando com Repositórios Remotos (GitHub)

- Criando um repositório no GitHub.
- Gerenciando repositórios remotos
- Enviando e recebendo alterações do repositório remoto.
- Criando Pull Requests

## 8. O arquivo .gitignore e .gitkeep

## 9. Entendendo GitFlow

## 10. Integração com Visual Studio Code



# Combinados

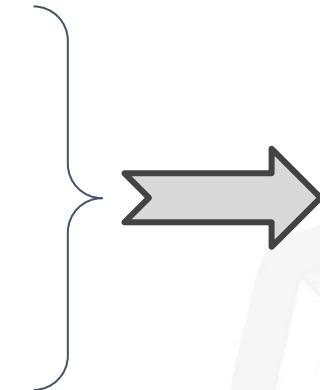
- Sempre às **8AM** estaremos online
- Notificações serão **desligadas**, mas sua contribuição é sempre bem-vinda
- Quando houver ADs, a explicação é **pausada**
- Evite **spamar** a mesma coisa no chat, moderadores estarão de olho
- Comentários discriminatórios e babacas receberão **banimento** sem aviso prévio



# Avisos

- Este é um curso de Git e GitHub básico/iniciante, tenha paciência
- Nosso foco é o básico bem feito

- Tem algo que não abordei?
- Tem algo que você explicaria diferente?
- Tem algo mais complexo que deixei passar?



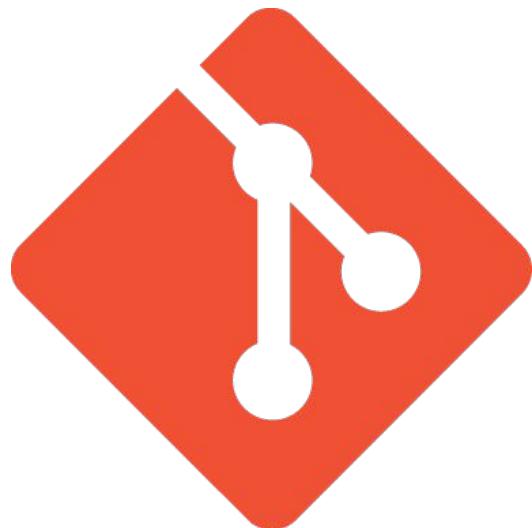
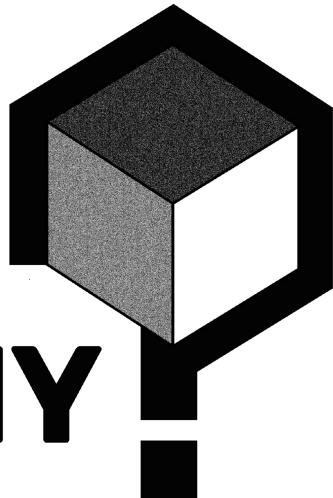
**Faça um  
vídeo/material**



# Git e GitHub

Uma introdução ao versionamento de código e trabalho colaborativo

TÉO  
ME WHY?



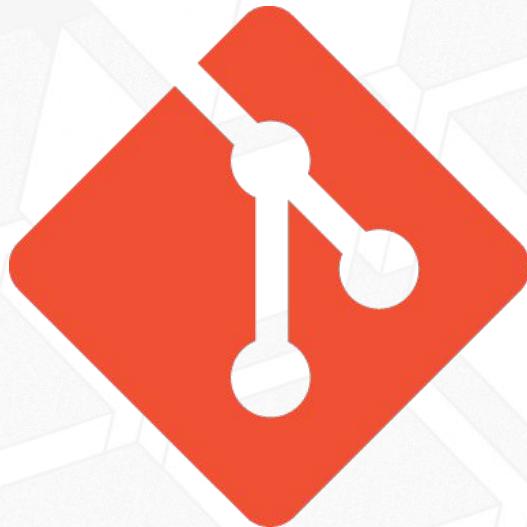
21/0/2025

Git e GitHub © 2025 by Téo Calvo is licensed under CC BY-NC-SA 4.0.  
To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>



# O problema

- Como lidar com diversas alterações em seu projeto?
- Como tomar decisões que possam ser reversíveis?
- Como trabalhar com muitos contribuidores?
- Como distribuir?



Git é um software para gerenciamento de versão para arquivos de texto no computador.

Criado por Linus Torvalds para melhorar o trabalho colaborativo em torno o Kernel do Linux.

É open-source e pode ser instalado em qualquer plataforma.



# Setup

## Para Windows

Face o download: <https://git-scm.com/downloads>

Realize a instalação: **torcedores calma!!!!!!**

## Para GNU/Linux

**Debian/Ubuntu:** *sudo apt install git*

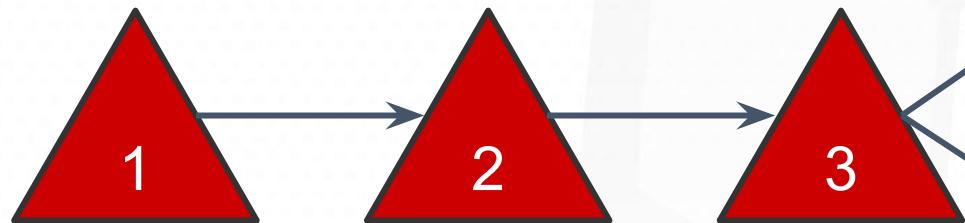
**Arch-based:** *yay -S git*

**Fedora:** *sudo yum install git*

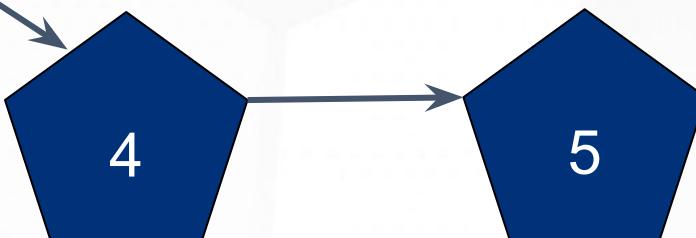
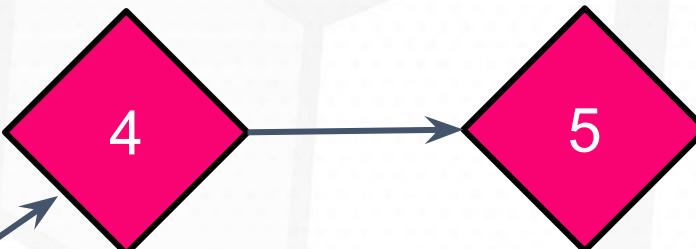


# Conceito

principal



secundária 01

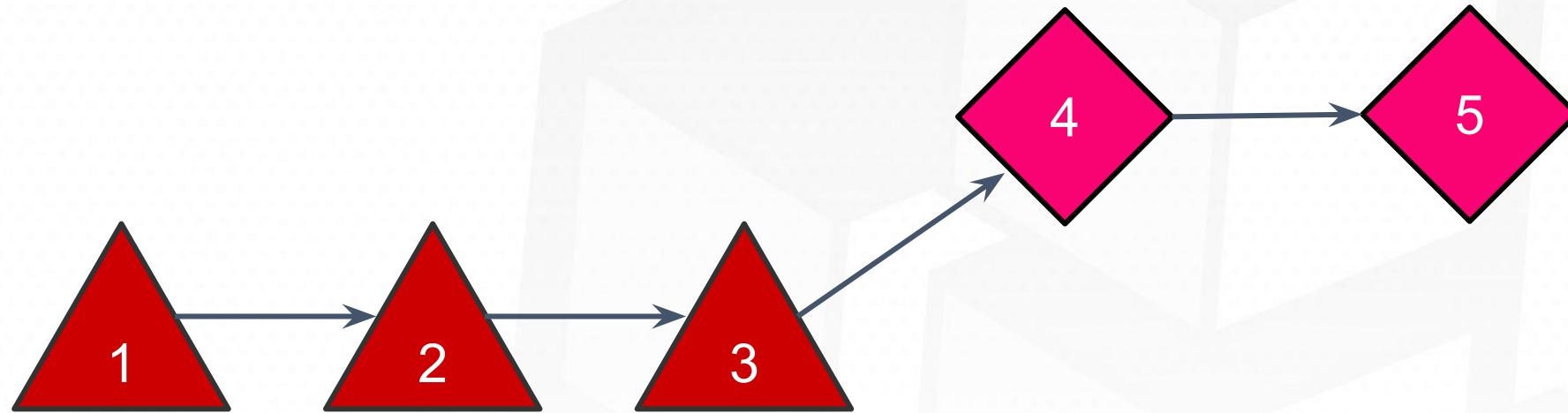


secundária 02





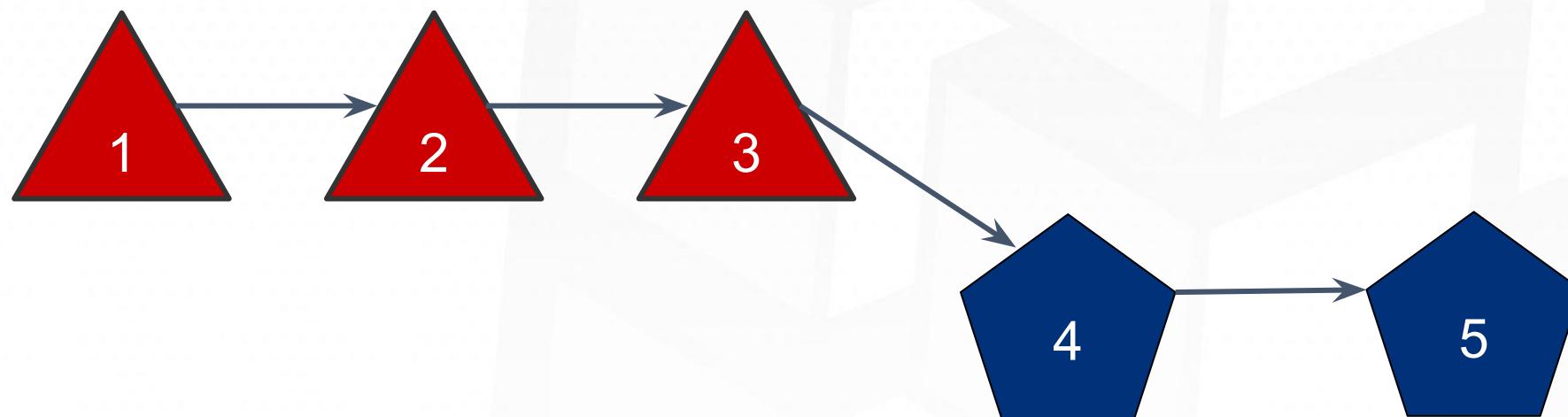
# Conceito



secundária 01



# Conceito

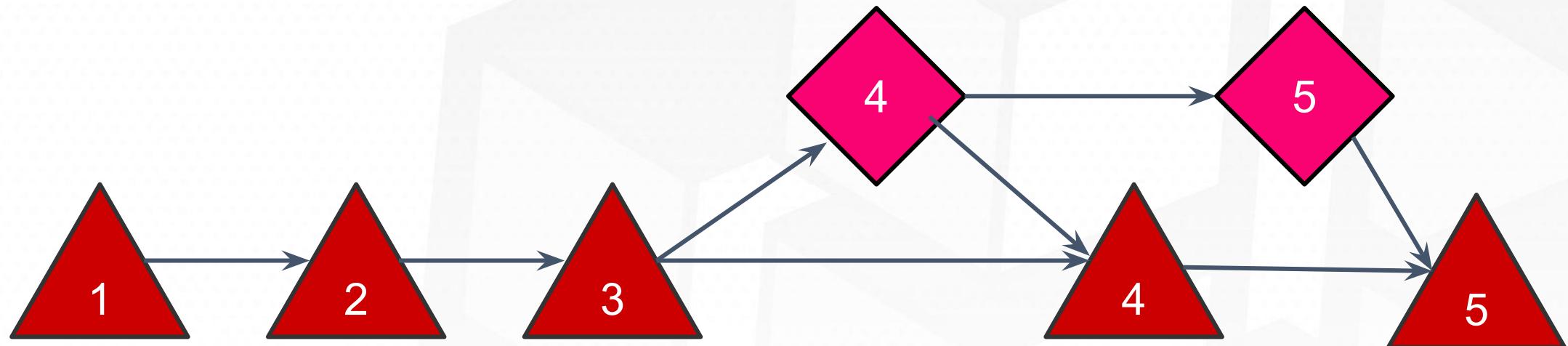


secundária 02



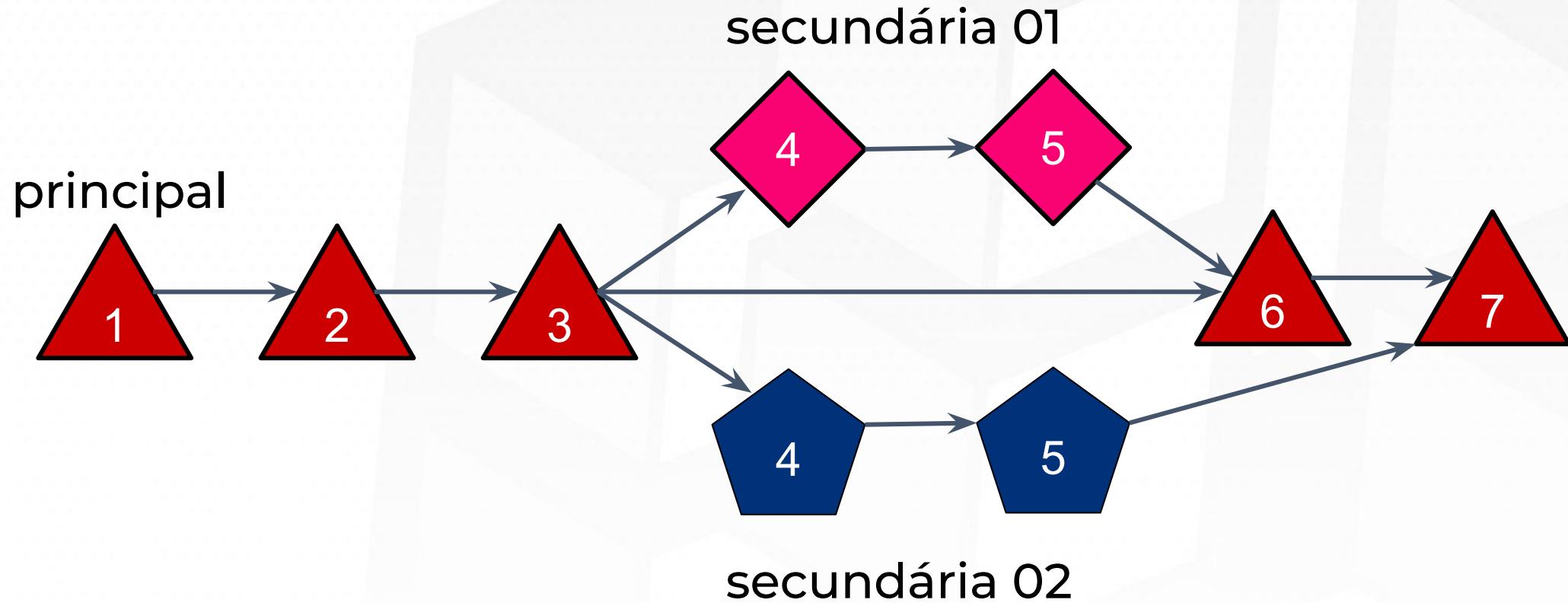
# Conceito

secundária 01





# Conceito





# Conceito

principal





# Antes de começar

Configurar nome e email para assinar commits:

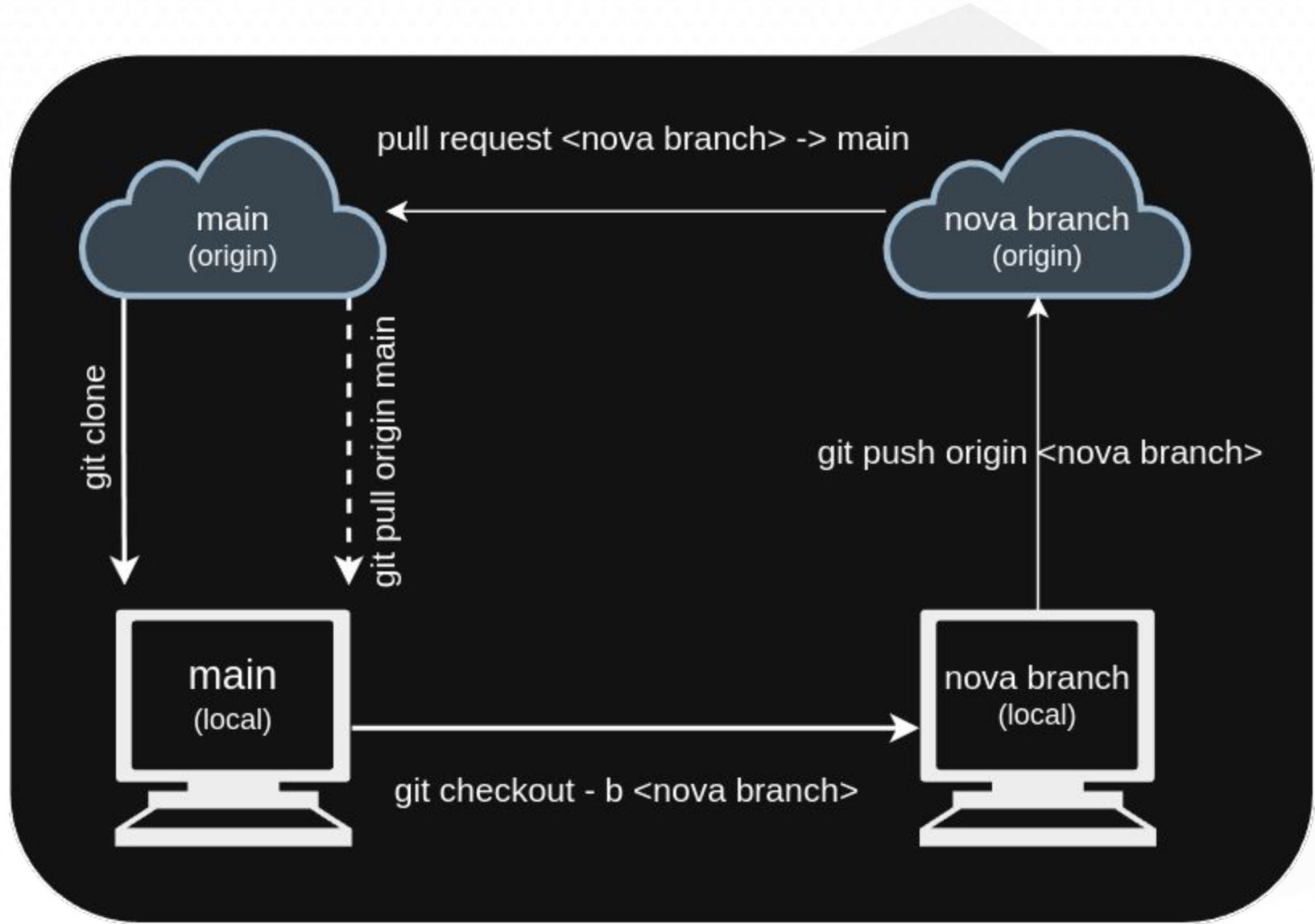
```
git config - -global user.name "Seu nome"
```

```
git config - -global user.email "seu@email"
```



# Comandos Básicos

Comando	Descrição
git init .	-> Cria novo repositório git
git status	-> Mostra o status da branch atual
git add <nome arquivo>	-> Manda arquivo para stage
git commit -m "minha mensagem"	-> Consolida o checkpoint "commit"
git log	-> Exibe histórico de commits
git reset	-> Retorna para unstage
git reset idCommit	-> Retorna para pos-commit

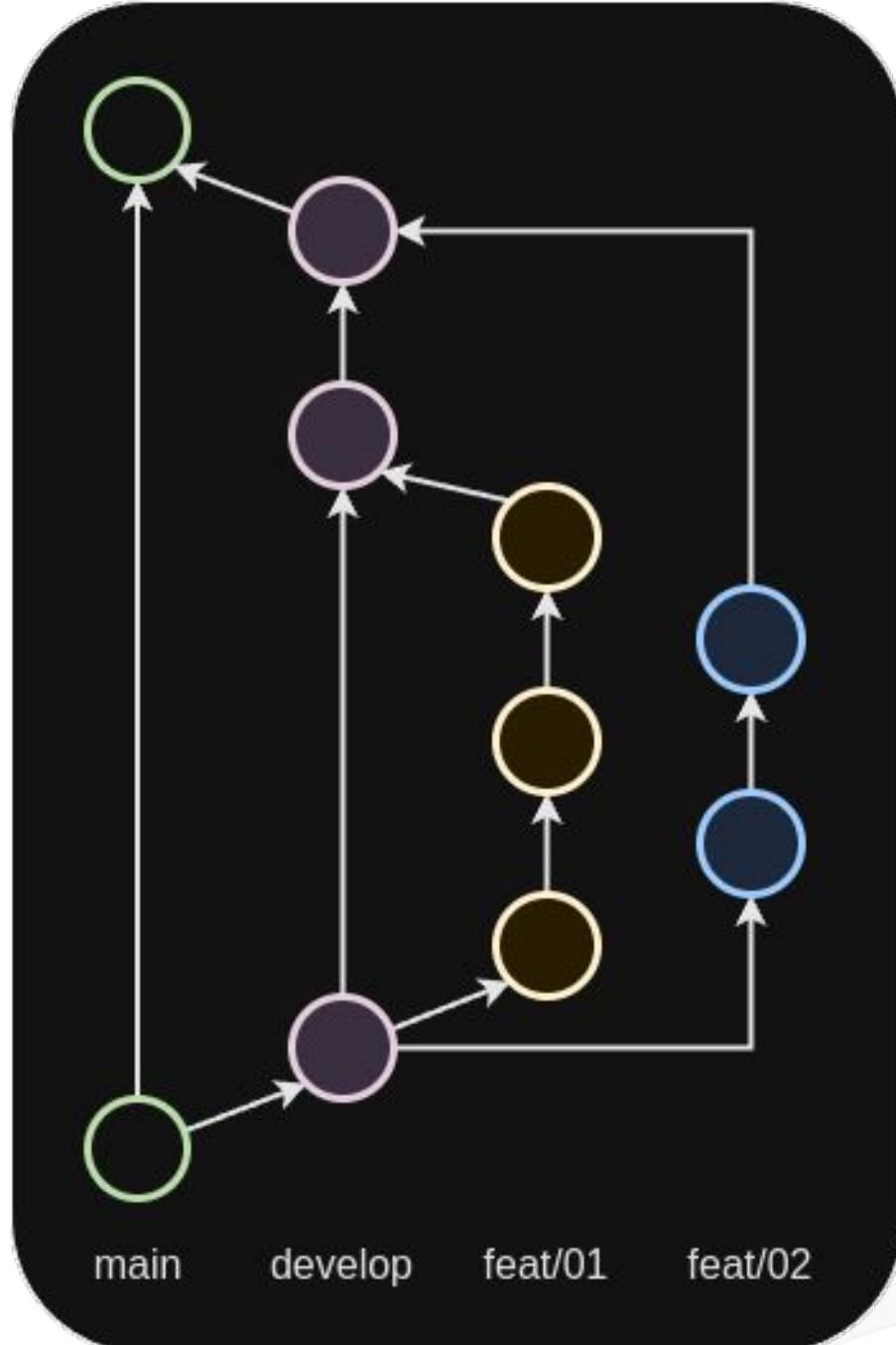




# Git-Flow

- checkout a partir da **develop**;
- pull-request para **develop**;
- release para **main**;

Assim, garantimos que o código está consistente antes de ir para produção;





“Big  
stories need  
**human**  
stakes”

Aaron Swartz

