



# IX concepar

CONGRESSO CIENTÍFICO DA  
REGIÃO CENTRO-OCIDENTAL  
DO PARANÁ

## GIT E GIT HUB

Douglas Nassif Roma Junior



/douglasjunior



/in/douglasjunior



douglasjunior.me



nassifroma@gmail.com



CENTRO UNIVERSITÁRIO  
**Integrado**

Repositório

<https://git.io/vpAuR>

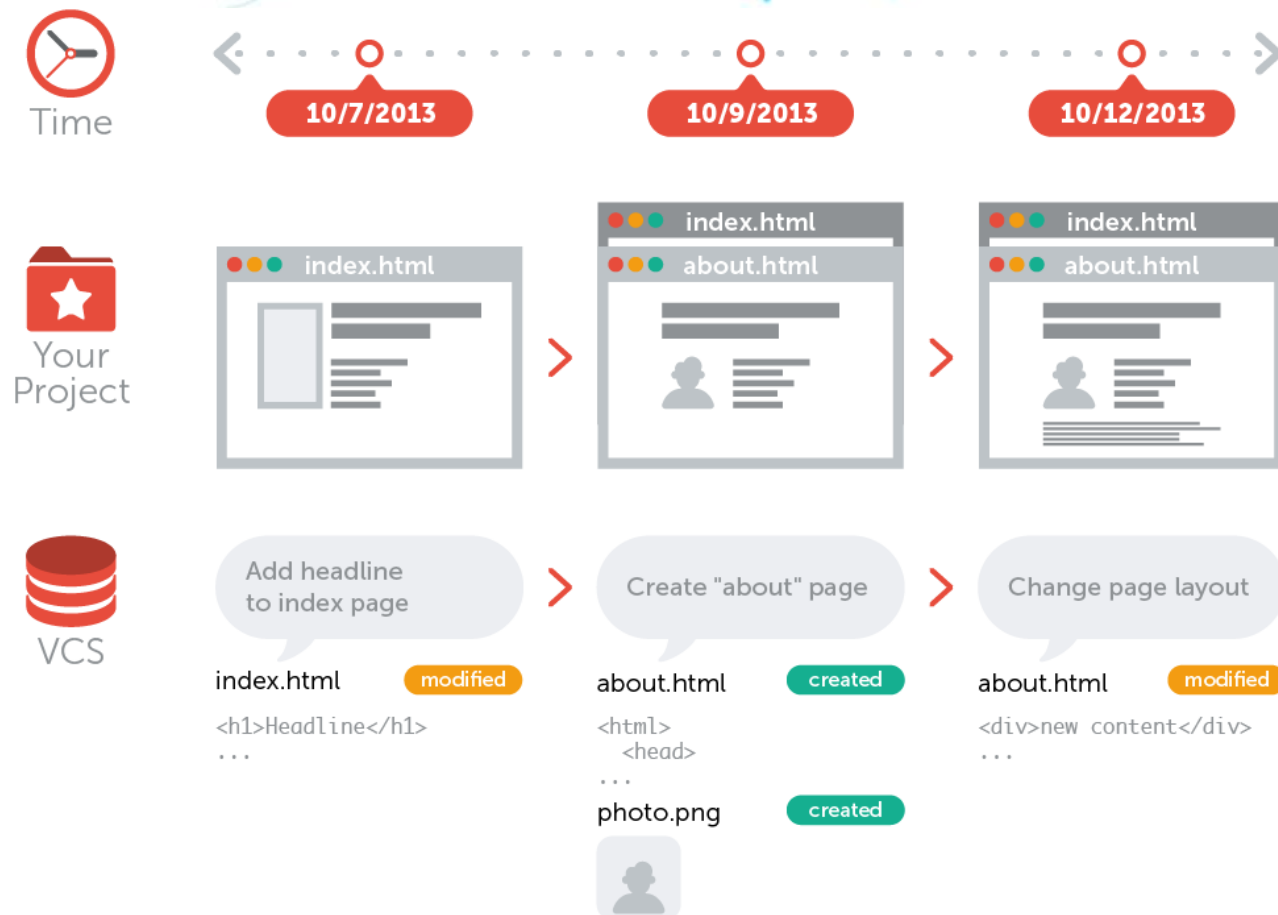
# Agenda

- Controle de versão – Como e por que?
- Ferramentas de controle de versão
- Git e GitHub
- Funcionalidades do Git
- Operações e comandos do Git



O que é controle de versão?

# Controle de versão



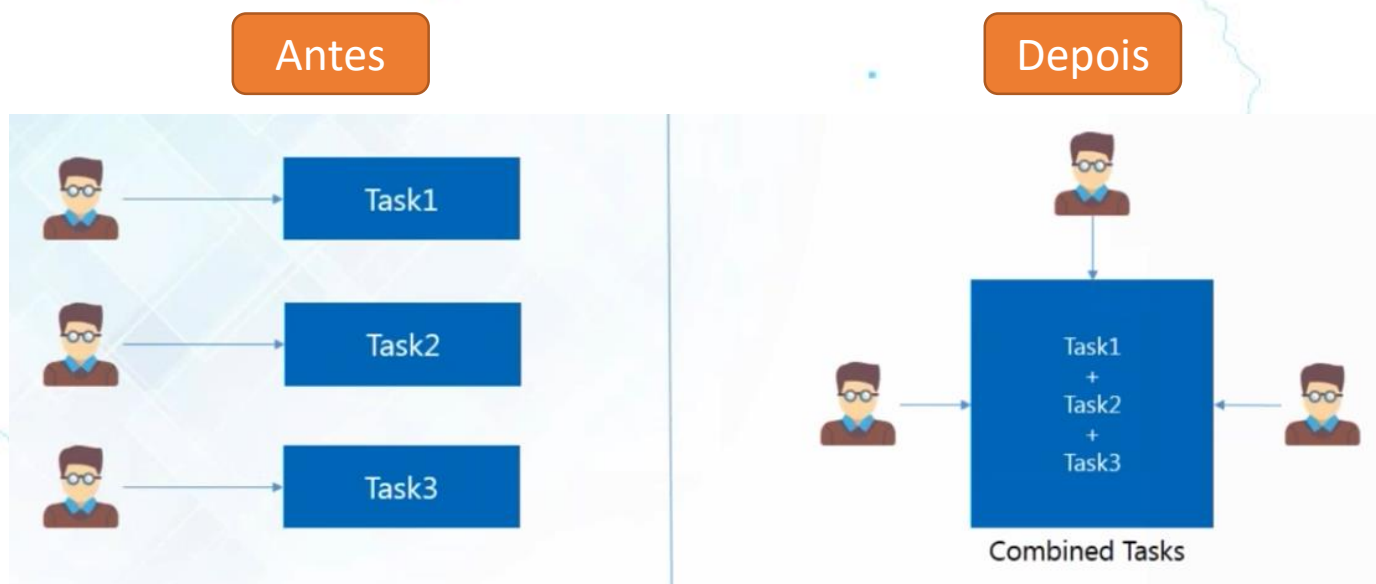
- “Controle de versão (VCS)” é o gerenciamento de mudança de documentos, programas de computadores, web sites e outras coleções de informações.
- Essas mudanças são conhecidas como “versões”.



**Por que precisamos de um  
controle de versão?**

# Controle de versão

- Colaboração



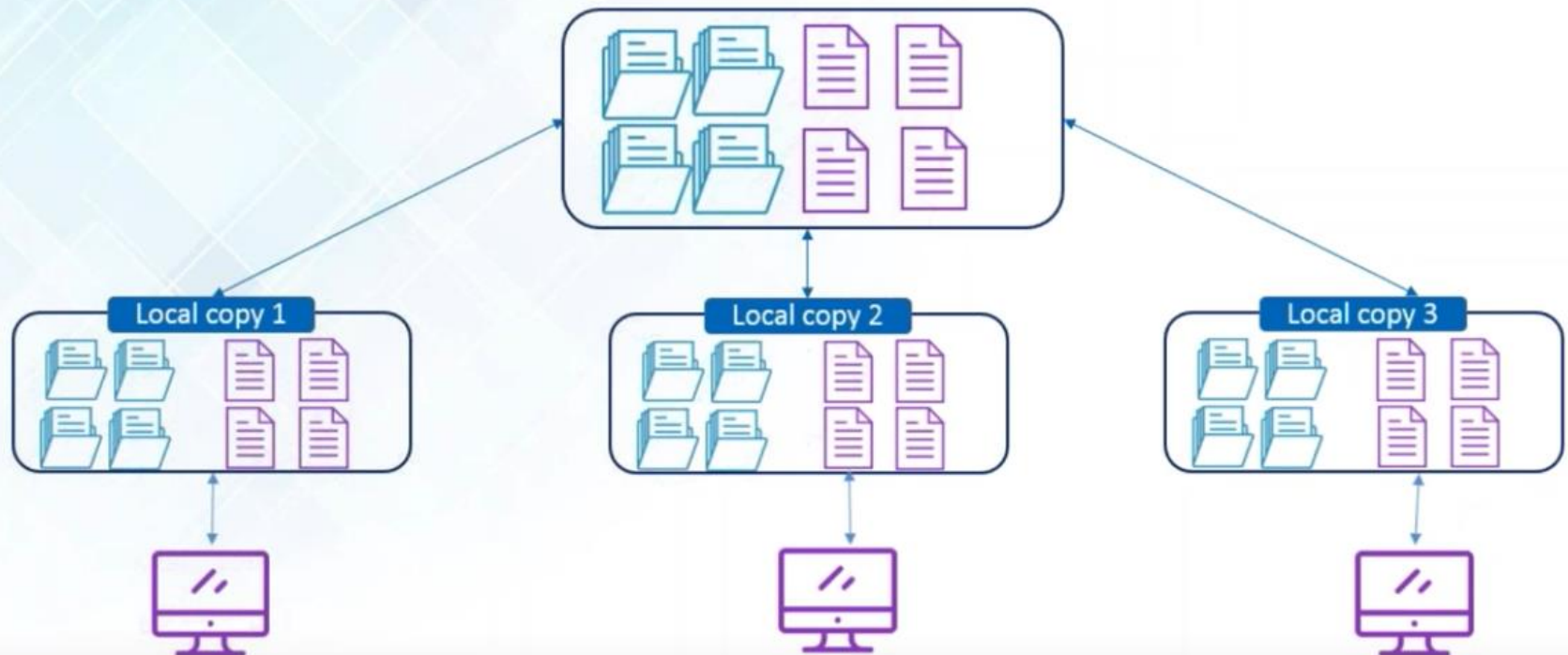
# Controle de versão

- Uma cópia de todas as versões são documentadas e armazenadas adequadamente.
- Versões são nomeadas com precisão.



# Controle de versão

- Caso o servidor principal exploda, você sempre terá uma cópia disponível do projeto em suas máquinas locais.





# Controle de versão

- Quando você faz uma alteração:
  - O VCS fornece a você uma descrição apropriada
  - O que realmente foi alterado
  - Quando foi alterado
- E então, você pode analisar como o seu projeto evoluiu ao longo das versões.





# Ferramentas de Controle de Versão

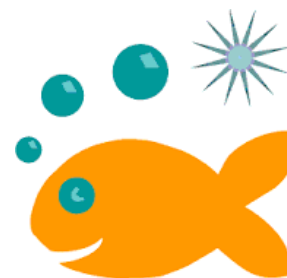
# Ferramentas de Controle de Versão



git



mercurial



CVS

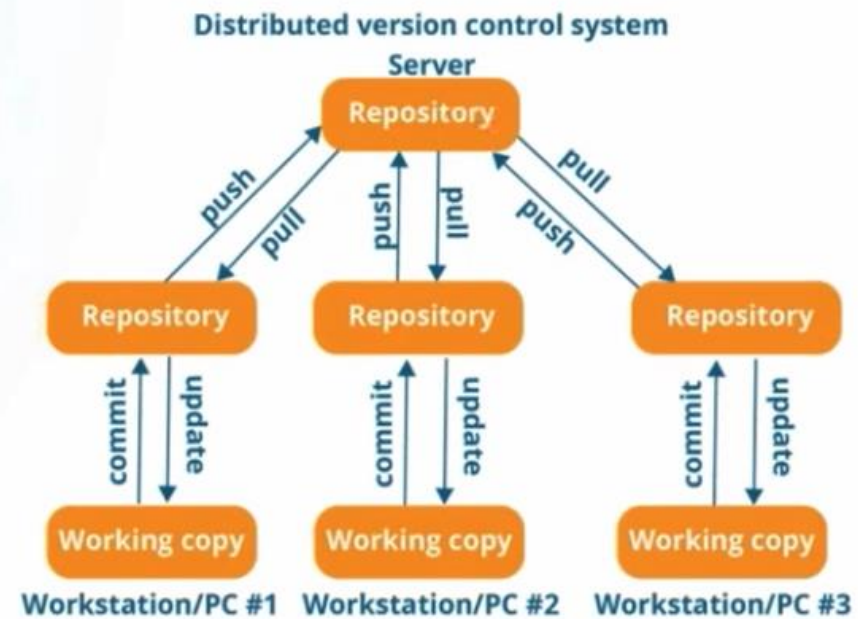
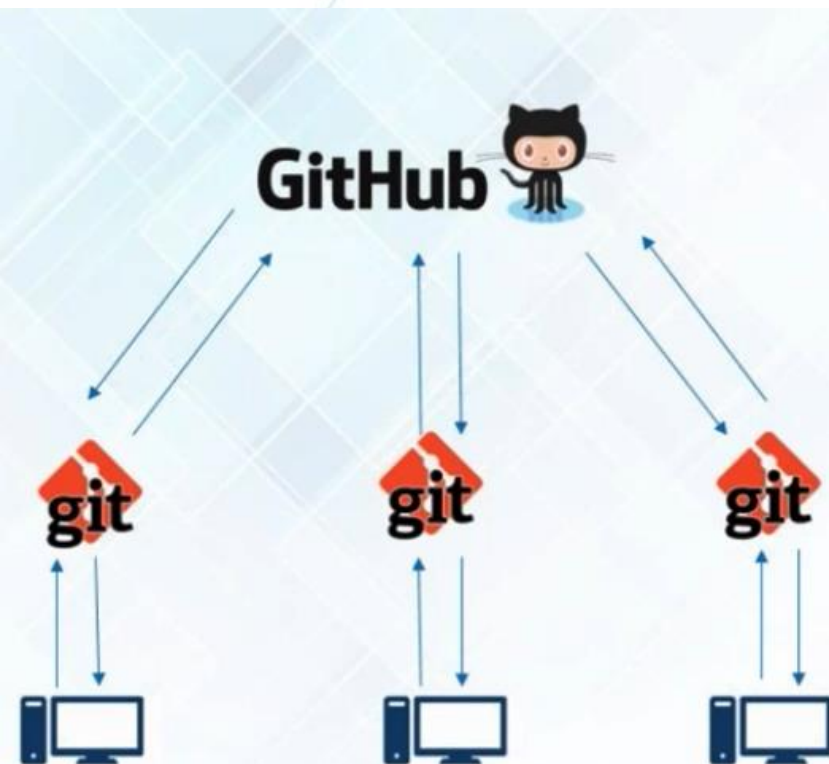
# Ferramentas de Controle de Versão



A light blue outline map of the state of Paraná, Brazil, is centered on the slide. Inside the map, there are approximately 15 small blue dots scattered across the territory. Along the northern and northeastern borders of the state, there are several 'X' marks, likely representing neighboring states or countries. The text 'Git e GitHub' is overlaid in the center of the map.

# Git e GitHub

# Git e GitHub

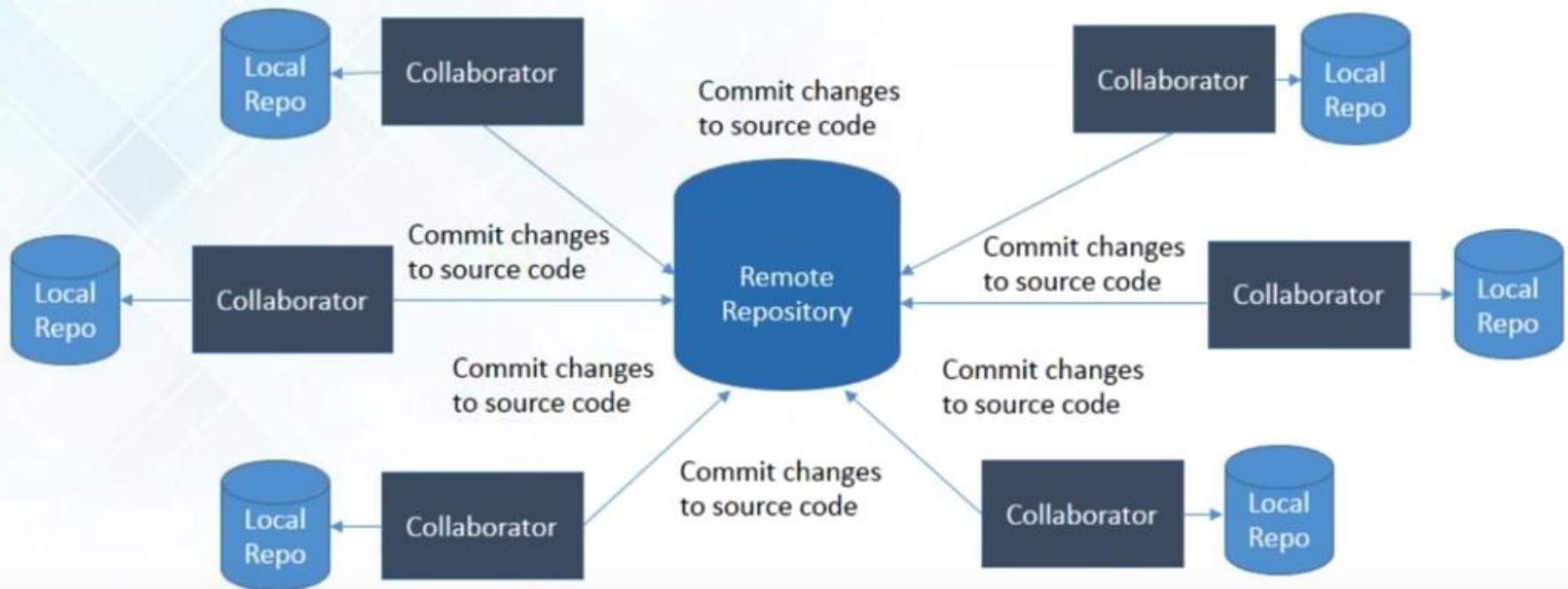


# O que é Git?



# Git

- Git é uma ferramenta de “Controle de Versão Distribuído” que suporta fluxos de trabalhos não lineares, e fornece garantia dos dados para ajudar na criação de softwares de qualidade.







# Funcionalidades do Git

# Git



Distributed



Compatible



Non-linear



Branching



Lightweight



Speed



Open Source



Reliable



Secure



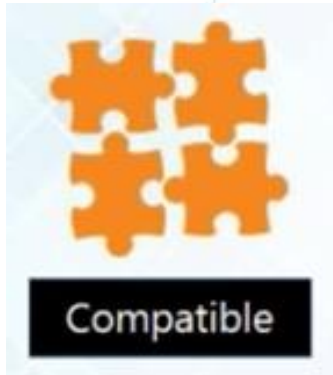
Economical

# Git



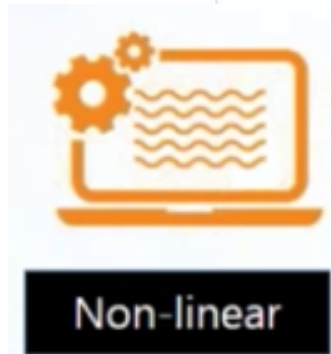
- Permite criação de código distribuído.
- Todo desenvolvedor possui uma cópia local do histórico de versões e mudanças.

# Git



- Compatível com sistemas operacionais (Windows, Linux e Mac) e protocolos (HTTP e SSH) existentes.
- Repositórios SVN e SVK podem ser acessados diretamente usando Git-SVN.

# Git



- Suporta desenvolvimento de software não-linear.
- Inclui várias técnicas para navegar e visualizar o histórico de forma não-linear.

# Git



## Branching

- Você leva apenas alguns segundos para criar ou unificar *branches* (ramos).
- A *branch master* contém o produto em versão funcional.

# Git



Lightweight

- Utiliza técnica de *lossless compression* (compressão sem perdas) para compactar os dados nos clientes.

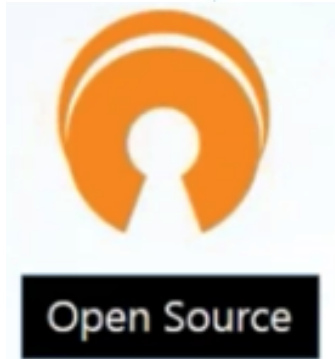
# Git



- Carregar dados de repositórios locais pode ser 100 vezes mais rápido do que em servidores remotos.
- Git é até 10 vezes mais rápido do que outros VCS.



# Git



- Você pode modificar o código fonte do Git de acordo com as suas necessidades.

# Git



Reliable

- Se um dos computadores falhar, os dados podem ser facilmente recuperados à partir de outros repositórios locais.

# Git



Secure

- Utiliza SHA1 para nomes e identificadores de objetos.
- Todos os arquivos e *commits* passam por verificação de integridade durante a gravação e leitura dos dados.

# Git



## Economical

- Distribuído sobre a licença GPL (General Public Licence).
- Todo o trabalho pesado é feito no lado do cliente, portanto você terá uma economia ao evitar a manutenção de servidores potentes.



# O que é um Repositório?

# Repositório

- Diretório ou espaço de armazenamento onde o seu projeto está localizado.
- Isso pode ser uma pasta em seu computador, ou um espaço de armazenamento no GitHub.

# Repositório

- Seu repositório pode conter:
  - Arquivos de código;
  - Imagens e vídeos;
  - Documentos;
  - Entre outros tipos de arquivos.

# Repositório

- Nós vamos falar de dois tipos de repositórios:
  1. Repositório Central
  2. Repositório Local



# Repositório

## Repositório Central

Normalmente localizado em um servidor remoto

Contem exclusivamente a pasta “.git”

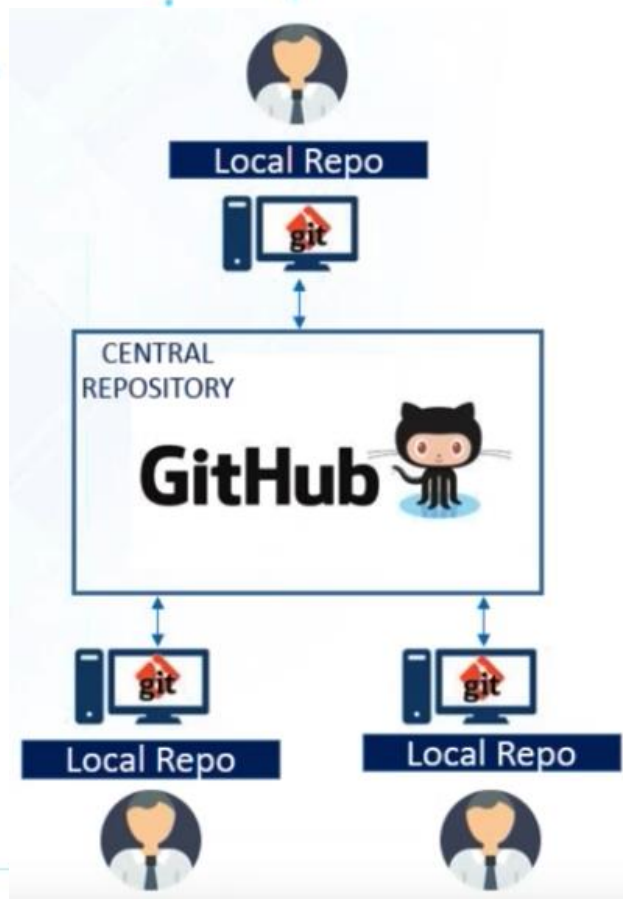
Utilizado pela equipe para compartilhar e trocar informações

## Repositório Local

Normalmente localizado na máquina local

Contem a pasta “.git” dentro do diretório do projeto

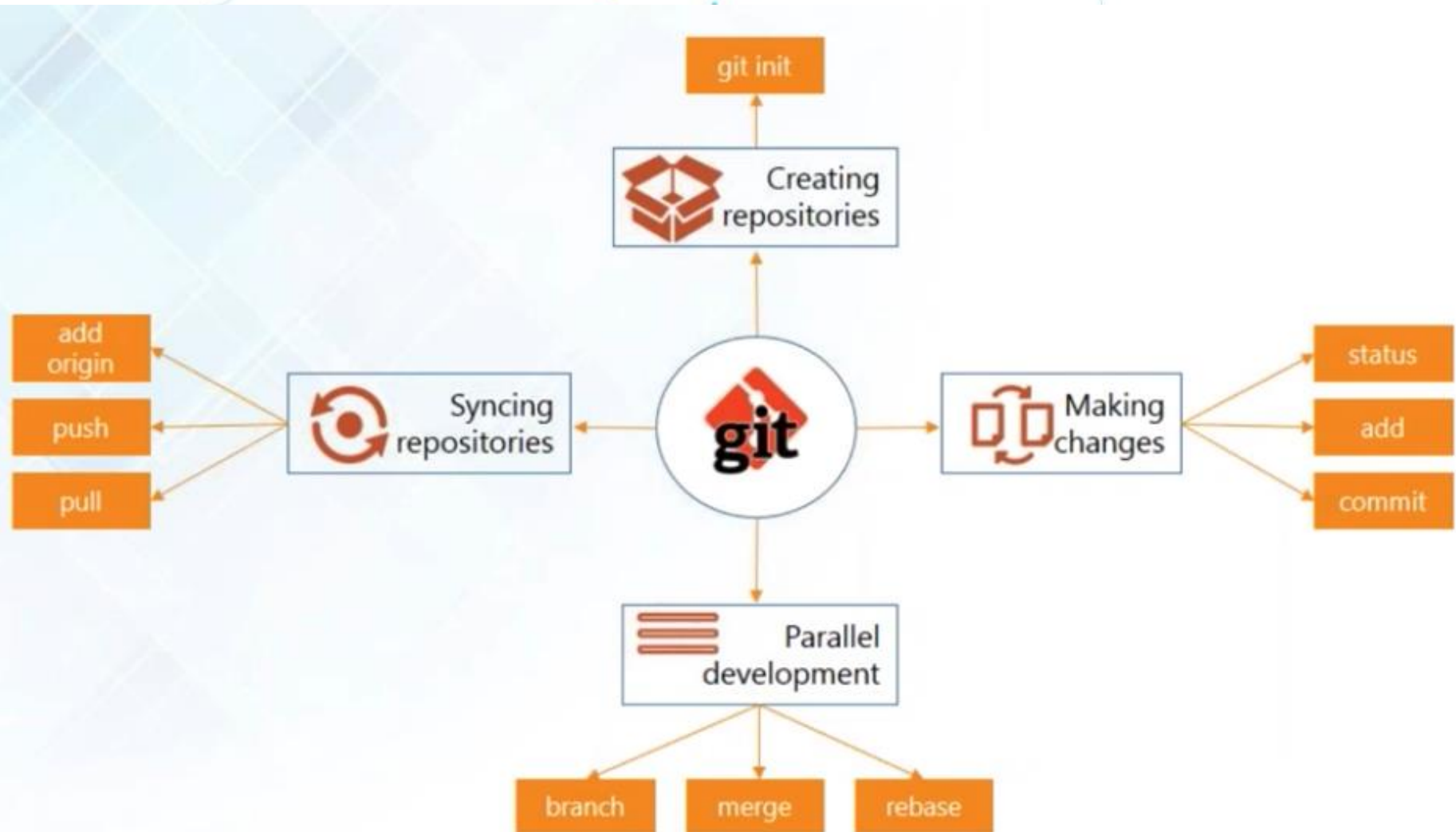
De uso exclusivo do administrador da máquina





# Operações e Comandos do Git

# Operações e Comandos



# Operações e Comandos

## git - guia prático

apenas um guia prático para começar com git. sem complicação ;)



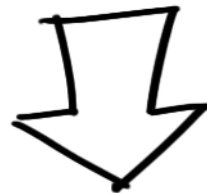
por Roger Dudler

créditos para @tfnico, @fhd and Namics

guia em english, deutsch, español, français, indonesian, italiano, nederlands, polski, русский, türkçe,

🇰🇷, 日本語, 中文, 한국어

por favor informe problemas em github



[http://rogerdudler.github.io/git-guide/index.pt\\_BR.html](http://rogerdudler.github.io/git-guide/index.pt_BR.html)



# IX concepar

CONGRESSO CIENTÍFICO DA  
REGIÃO CENTRO-OCIDENTAL  
DO PARANÁ

## Obrigado

Douglas Nassif Roma Junior



/douglasjunior



/in/douglasjunior



douglasjunior.me



nassifroma@gmail.com



CENTRO UNIVERSITÁRIO  
**Integrado**