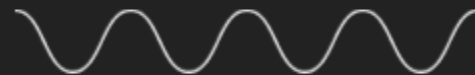


# CONTROLE DE VERSÃO, GIT E GITHUB





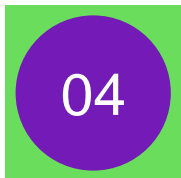
Controle de versão



Git



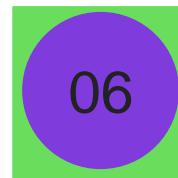
Terminal



Instalação Git



GitHub



Git comandos



# Controle de versão, git e github

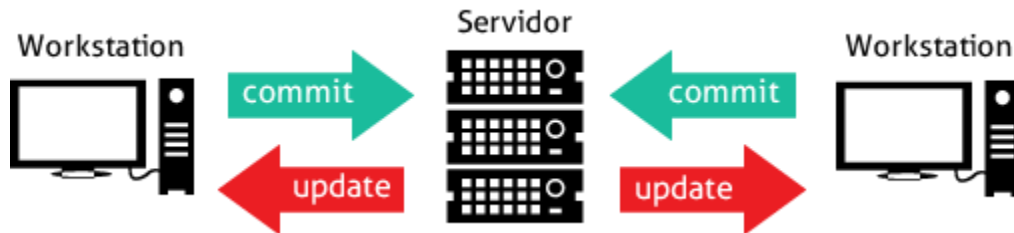
(Introdução)

# Controle de versão

---

O controle de versão consiste basicamente em um **sistema que permite registrar alterações** feitas no desenvolvimento de um software.

É a partir dele que toda a equipe envolvida no projeto têm acesso ao **histórico das versões anteriores** do software, podendo **recuperar uma versão específica** ou **compreender quais mudanças foram feitas** por outras pessoas.



# Git

---



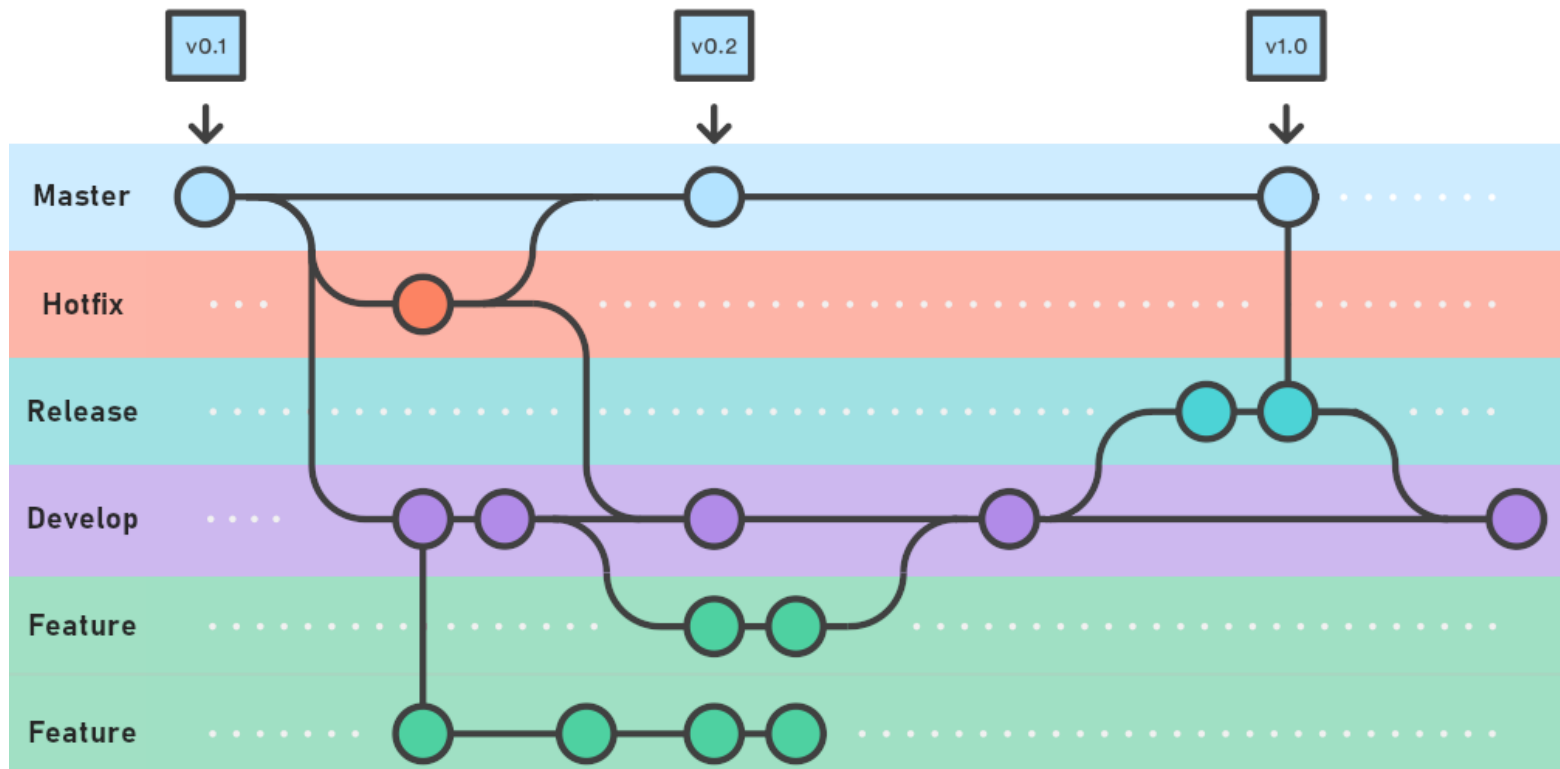
Git é um **sistema de controle de versões**, usado principalmente no desenvolvimento de software.

O Git é um **software livre** e foi inicialmente projetado e desenvolvido por **Linus Torvalds** para o desenvolvimento do kernel Linux.

Desde seu nascimento em 2005, Git evoluiu e amadureceu para ser fácil de usar. Ele é incrivelmente rápido, é muito eficiente com projetos grandes, e ele tem um incrível sistema de ramos (branches) para desenvolvimento não linear.



# Git





# Terminal

---

Basicamente, terminal é aquela **famosa tela preta** na qual você **digita comandos para dar instruções para um computador**. Ou seja, ele serve para você **executar tarefas no computador sem utilizar a interface gráfica**, com pastinhas e ícones, ou o bom e velho mouse. **Todos os comandos são executados através de digitação de texto puro.**

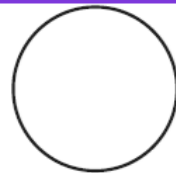
# Terminal - comandos básicos

---

<b>ls (macOS/Linux)</b> <b>dir (Windows)</b>	lista todos os arquivos presentes no diretório atual (macOS/Linux)
<b>mkdir nome-da-pasta</b>	cria uma nova pasta
<b>cd nome-da-pasta</b>	navega para a pasta especificada (exemplo: cd documentos)
<b>cd ..</b>	sobe um nível de pasta
<b>touch nome-do-arquivo</b> <b>dir &gt; nome-do-arquivo</b>	cria um novo arquivo
<b>clear</b>	limpa todas as informações do terminal
<b>&lt;nome do programa&gt; --version</b>	Exibe versão instalada



# Instalação Git





# Instalação Git

<http://git-scm.com/download/win>

## Baixando Git



Você está baixando a versão mais recente ( **2.33.0** ) de **64 bits** do **Git para Windows** . Esta é a **compilação mantida** mais recente . Foi lançado há **cerca de 2 meses** , em 2021-08-24.

[Clique aqui para baixar manualmente](#)

### Outros downloads do Git para Windows

Configuração do Git para Windows

[Git de 32 bits para instalação do Windows](#) .

[Git de 64 bits para instalação do Windows](#) .

Git para Windows Portable ("edição thumbdrive")

[Git de 32 bits para Windows portátil](#) .

[Git de 64 bits para Windows portátil](#) .

A versão atual do código-fonte é **2.33.0** . Se você quiser a versão mais recente, pode compilá-la a partir [do código-fonte](#) .



# GitHub

GitHub é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git.

Ele permite que **qualquer pessoa cadastrada** na plataforma **contribua em projetos** privados e/ou de código-fonte aberto (Open Source) de **qualquer lugar do mundo**.





# Git - configuração inicial

---

1. Crie conta GitHub: <https://github.com/login>
2. Instale o Git na sua máquina
3. Configure suas informações (nome e email Github)
  - a. `git config --global user.name "Seu Nome"`
  - b. `git config --global user.email "Seu Email"`
4. Verifique suas informações
  - a. `git config --list` OU
  - b. `git config user.name` OU
  - c. `git config user.email`

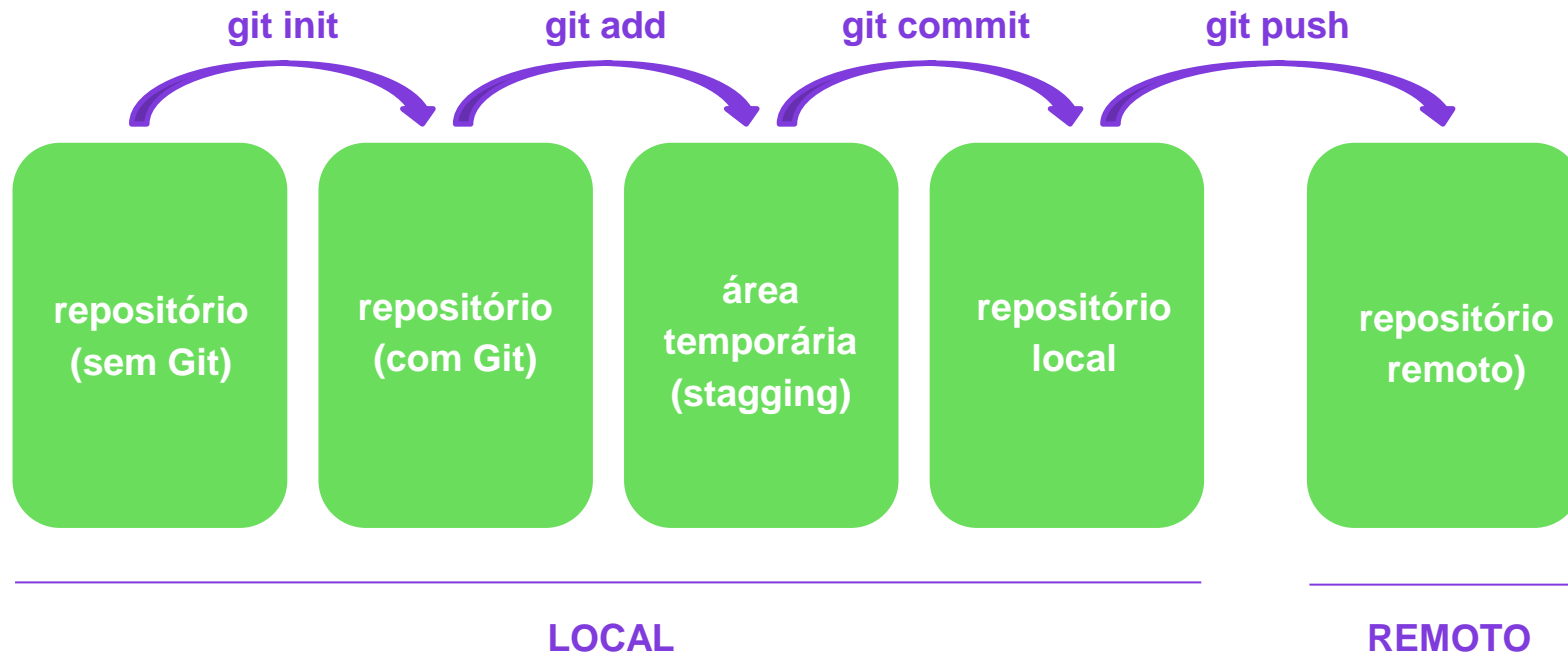
# Git - comandos básicos

---

<b>git init</b>	inicializa o git no repositório local
<b>git add nome-do-arquivo ou .</b>	adiciona um arquivo modificado ao staging (área temporária)
<b>git status</b>	mostra os status dos arquivos modificados
<b>git commit -m "mensagem"</b>	cria um commit
<b>git pull</b>	puxa as atualizações mais recente (remoto -> local)
<b>git push</b>	envia as atualizações mais recentes (local -> remoto)
<b>git remote add origin &lt;caminho&gt;</b>	adiciona o seu repositório local ao remoto
<b>git checkout -- &lt;nome-arquivo&gt;</b>	descarta as alterações locais do arquivo informado

# Git - comandos básicos

---





# Dicas importantes

---

1. As mensagens do commit devem ser claras e sucintas, descrevendo bem a alteração
2. Um push só funciona se houver um commit feito!
3. Não se desespere, Git pode parecer complicado no começo mas depois você faz de olhos fechados (ou quase isso haha)

VAMOS PRATICAR?





# Exercícios

---

1. Apresentação
  - a. Crie um repositório localmente e inicialize o git
  - b. Adicione um arquivo markdown chamado README.md com seu nome e prato favorito e faça um commit
  - c. Adicione uma curiosidade sobre você e faça outro commit
  - d. Publique o repositório no seu GitHub

[Guia básico sobre markdown](#)



# Exercícios

---

## 2. Fork

- a. Faça um fork do repositório.
- b. Clone o repositório para a sua máquina.
- c. Crie uma nova branch com seu nome (exemplo: `git checkout -b jenifer-lp`).
- d. Faça commits com a resolução dos exercícios.
- e. Atualize seu repositório remoto.

