

git - guia prático

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

Tweet

por Roger Dudler

créditos para @tfnico, @fhd and Namics

english, deutsch, español, français, indonesian, italiano, nederlands, polski, русский,

မြန်မာ, 日本語, 中文, 한국어

por favor informe problemas em github

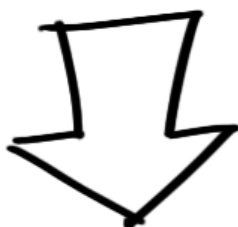


Are You a Front-End Developer?

by Roger Dudler, Author of the Git Simple Guide

Try Frontify

Now Free with
Github Integration!



instalação

Baixe o git para OSX

Baixe o git para Windows

Baixe o git para Linux

criando um novo repositório

crie uma nova pasta, abra-a e execute o comando

```
git init
```

para criar um novo repositório.

obtenha um repositório

crie uma cópia de trabalho em um repositório local executando o comando

```
git clone /caminho/para/o/repositório
```

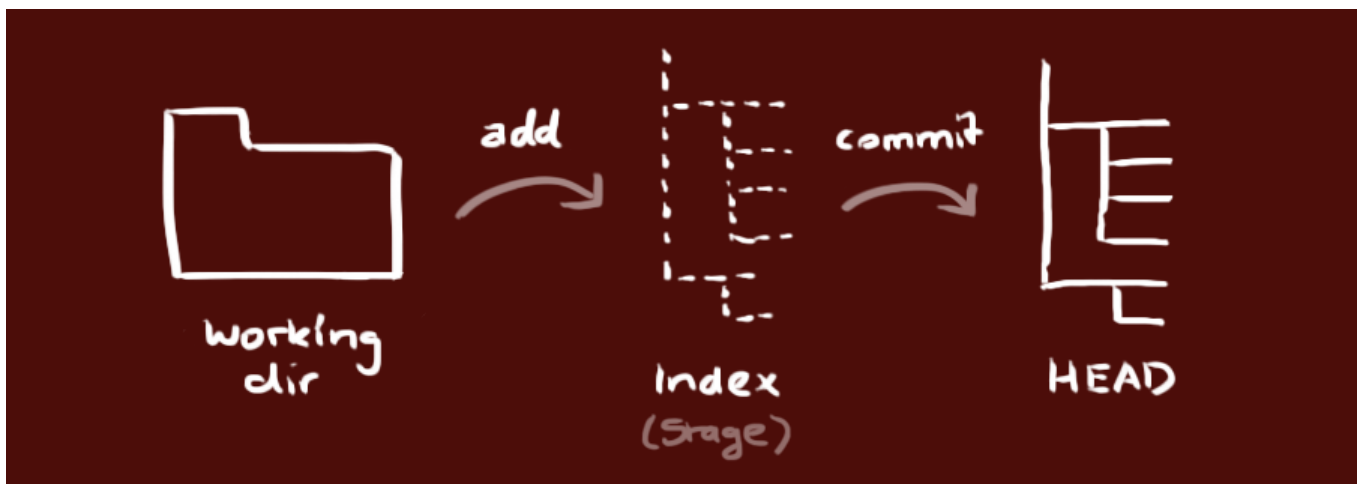
quando usar um servidor remoto, seu comando será

```
git clone
```

```
usuário@servidor:/caminho/para/o/repositório
```

fluxo de trabalho

seus repositórios locais consistem em três "árvores" mantidas pelo git. a primeira delas é sua **Working Directory** que contém os arquivos vigentes. a segunda **Index** que funciona como uma área temporária e finalmente a **HEAD** que aponta para o último *commit* (confirmação) que você fez.



adicionar & confirmar

Você pode propor mudanças (adicioná-las ao **Index**) usando

```
git add <arquivo>
```

```
git add *
```

Este é o primeiro passo no fluxo de trabalho básico do git. Para realmente confirmar estas mudanças (isto é, fazer um *commit*), use

```
git commit -m "comentários das alterações"
```

Agora o arquivo é enviado para o **HEAD**, mas ainda não para o repositório remoto.

enviando alterações

Suas alterações agora estão no **HEAD** da sua cópia de trabalho local.

Para enviar estas alterações ao seu repositório remoto, execute

```
git push origin master
```

Altere *master* para qualquer ramo (*branch*) desejado, enviando suas alterações para ele.

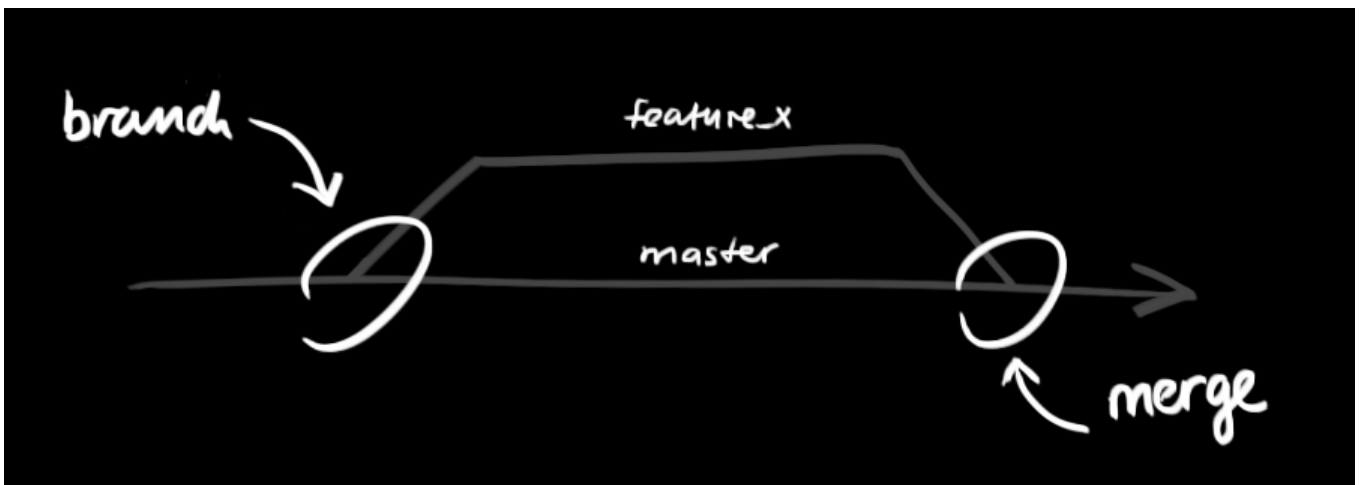
Se você não clonou um repositório existente e quer conectar seu repositório a um servidor remoto, você deve adicioná-lo com

```
git remote add origin <servidor>
```

Agora você é capaz de enviar suas alterações para o servidor remoto selecionado.

ramificando

Branches ("ramos") são utilizados para desenvolver funcionalidades isoladas umas das outras. O branch *master* é o branch "padrão" quando você cria um repositório. Use outros branches para desenvolver e mescle-os (*merge*) ao branch master após a conclusão.



crie um novo branch chamado "funcionalidade_x" e selecione-o usando

```
git checkout -b funcionalidade_x
```

retorne para o master usando

```
git checkout master
```

e remova o branch da seguinte forma

```
git branch -d funcionalidade_x
```

um branch *não está disponível a outros* a menos que você envie o

branch para seu repositório remoto

```
git push origin <funcionalidade_x>
```

atualizar & mesclar

para atualizar seu repositório local com a mais nova versão, execute

```
git pull
```

na sua pasta de trabalho para *obter e fazer merge* (mesclar) alterações remotas.

para fazer merge de um outro branch ao seu branch ativo (ex. master),

use

```
git merge <branch>
```

em ambos os casos o git tenta fazer o merge das alterações automaticamente. Infelizmente, isto nem sempre é possível e resulta em

conflitos. Você é responsável por fazer o merge estes *conflitos* manualmente editando os arquivos exibidos pelo git. Depois de alterar, você precisa marcá-los como merged com

```
git add <arquivo>
```

antes de fazer o merge das alterações, você pode também pré-visualizá-

as usando

```
git diff <branch origem> <branch destino>
```

rotulando

é recomendado criar rótulos para releases de software. Este é um conhecido conceito, que também existe no SVN. Você pode criar um novo rótulo chamado *1.0.0* executando o comando

```
git tag 1.0.0 1b2e1d63ff
```

o *1b2e1d63ff* representa os 10 primeiros caracteres do id de commit que você quer referenciar com seu rótulo. Você pode obter o id de commit

com

```
git log
```

você pode também usar menos caracteres do id de commit, ele somente precisa ser único.

sobrescrever alterações locais

No caso de você ter feito algo errado (que seguramente nunca acontece ;)) você pode sobrescrever as alterações locais usando o commando

```
git checkout -- <arquivo>
```

isto substitui as alterações na sua árvore de trabalho com o conteúdo mais recente no HEAD. Alterações já adicionadas ao index, bem como novos arquivos serão mantidos.

Se ao invés disso você deseja remover todas as alterações e commits locais, recupere o histórico mais recente do servidor e aponte para seu

branch master local desta forma

```
git fetch origin
```

```
git reset --hard origin/master
```

dicas úteis

Interface gráfica padrão

`gitk`

usar saídas do git coloridas

`git config color.ui true`

exibir log em apenas uma linha por commit

`git config format.pretty oneline`

fazer inclusões interativas

`git add -i`

recursos & links

clientes gráficos

GitX (L) (OSX, código aberto)

Tower (OSX)

Source Tree (OSX, gratuito)

GitHub for Mac (OSX, gratuito)

GitBox (OSX)

guias

Livro da comunidade Git

Pro Git

Pense como um git

Ajuda do GitHub

Um guia visual do Git

Clarify

Building websites? Working with Designers? Try Clarify.
A Project by Roger Dudler, Author of The Git Simple Guide



comentários

Sponsored Links

How this app teaches you a language in 3 weeks!

Babbel

Brilliant Device Turns Ordinary Phones into DSLR-like Camera

HDZoom360

56 of the Worst Cars of All Time

Carophile

6 Jobs That Will Be Gone in 10 Years

BleuBloom.com

These 20 Countries Have The Highest Employment Rates

WomensArticle.com

Best Tracking Device Ever is Selling Like Crazy

TrackR Bravo

187 Comments

git - the simple guide

 Login ▾

 Recommend 92

 Share

Sort by Newest ▾



Join the discussion...

Claudio Junior Fabião • 7 days ago

Demais! Bom trabalho para quem está começando no Git! ;)

^ | ▾ • Reply • Share ›

Rafa Costa • 16 days ago

Nice, man !

^ | ▾ • Reply • Share ›

djmarksouza • 21 days ago

artigo excelente, muito obrigado ajudou muito.

^ | ▾ • Reply • Share ›

Josh Chip • 23 days ago

pra mim vc e o mais pika danado

^ | v · Reply · Share ›

Willer Reis · a month ago

Sensacional!!!! Muito útil de forma prática e simples!

Como um amigo disse: Simples e Direto.

Parabéns

^ | v · Reply · Share ›

wilian cesar · a month ago

Muito bom!!!!!!!!!!!!

^ | v · Reply · Share ›

Samuelson Brito · 2 months ago

Utilidade pública.

1 ^ | v · Reply · Share ›

Cerebro Vasconcelos → Samuelson Brito · a month ago

Muita utilidade

1 ^ | v · Reply · Share ›

Wong · 2 months ago

Is change necessary in "git checkout master" by "git checkout -b master"?

^ | v · Reply · Share ›

Alex André Crocetti · 2 months ago

Simples e direto.

^ | v · Reply · Share ›

David Almeida · 3 months ago

Mt obg ajudou bastante com uma duvida

^ | v · Reply · Share ›

Marcos Camargo · 3 months ago

Muito bom!

^ | v · Reply · Share ›

adilmar · 3 months ago

Ótimo tutorial :D

^ | v · Reply · Share ›

Fabício Augusto · 3 months ago

Thank you! :)

^ | v · Reply · Share ›

Dorival Cardozo · 4 months ago

Para quem já conhece o GIT serve como um excelente lembrete, mas para quem esta começando ainda é bem confuso e não ajuda muiito.

3 ^ | v · Reply · Share ›



Leandro · 4 months ago

Otimas explicações

^ | v · Reply · Share ›

Guibson · 5 months ago

Top tutorial!! Muito claro, objetivo!! Valeu, meu caro!!

^ | v · Reply · Share ›

Wanderson Silva · 6 months ago

Amo esse site, Usabilidade 100% <3

^ | v · Reply · Share ›

Gabriel L. Lopes · 6 months ago

Gabriel Lopes · 6 months ago

Show!!

^ | v · Reply · Share ›

Vinícius Carneiro · 6 months ago

Top!!!!

^ | v · Reply · Share ›



geovane · 7 months ago

Esse ta salvo nas minhas referências

^ | v · Reply · Share ›

Josiel Faleiros · 7 months ago

Ajudou bastante!

^ | v · Reply · Share ›

adrisson galvao · 8 months ago

Muito bom cara parabéns. Muito Obrigado

^ | v · Reply · Share ›

Mylena Mariana · 8 months ago

Muiito obrigada!!!

^ | v · Reply · Share ›

Saulo Calixto · 8 months ago

Muito bom, obrigado!

^ | v · Reply · Share ›

SenHor · 8 months ago

NICE! :)

1 ^ | v · Reply · Share ›

Junior Oliveira · 8 months ago

Muito Obrigado

^ | v · Reply · Share ›

Pedro Henrique Ferreira Fonsec · 8 months ago

Estava precisando de um tutorial simplificado como esse. Vlw d+!!

^ | v · Reply · Share ›

Igor Guilherme · 9 months ago

Muito obrigado! Simples e objetivo, parabéns!

^ | v · Reply · Share ›

italochesley · 10 months ago

sempre que eu preciso fazer algum commit recorro a esse tutorial...muito foda!

Parabéns!

1 ^ | v · Reply · Share ›

Erikson Magno · 10 months ago

Muito bom. Valeu mesmo

^ | v · Reply · Share ›

John · 10 months ago

Obrigado! Bem detalhado.

^ | v · Reply · Share ›

Nayara Valadares · 10 months ago

Muito bom!! obrigada :D

^ | v · Reply · Share ›

Sor Dantas, o Magro · 10 months ago

Muito Bom!

^ | v · Reply · Share ›

Obrigado!

^ | v · Reply · Share ›

Jackson · a year ago

Parabéns pelo tutorial muito bom!

Instalei o GitLab no servidor, sou iniciante, criei um repository, como faço para jogar meus commits em uma pasta /var/www/meuprojeto, esta pasta fica dentro do servidor apache, assim que eu fizer a alteração eu possa acessar os arquivos através da url dele www.meudominio.com/meuprojeto.

Alguem pode me dar um help?

^ | v · Reply · Share ›

Yves Silva Romeiro · a year ago

Voce é o cara me ajudou muito !!!!!

^ | v · Reply · Share ›

Eris R. Jr · a year ago

Muito bacana.

*O link para download/instruções de instalação no Linux agora é:
<https://git-scm.com/download...>

^ | v · Reply · Share ›



Anderson · a year ago

Melhor tutorial de git da internet, sem dúvida! Valeu!

^ | v · Reply · Share ›



Te · a year ago

simplesmente perfeito! não ajudou só eu, como todos do time de desenvolvimento da empresa favoritaram..rsrs

^ | v · Reply · Share ›



Jones · a year ago

Simplesmente demais! Valeu muito pela forma clara e método de fazer armazenar as informações de uma vez por todas..... OBRIGADU

^ | v · Reply · Share ›

Marcelo Candido · a year ago

Excelente!

^ | v · Reply · Share ›



beatz · a year ago

Perfeito, muito obrigado

^ | v · Reply · Share ›

Jader Barros Pereira · a year ago

Perfeito, lindo, inestimável e incomensurável o valor deste guia. Meus parabéns ao autor :)

^ | v · Reply · Share ›

TiagoCeo · a year ago

Esse guia salvou, eu tava usando o cliente TortoiseGit legal, mais que saber, vou usar é na linha de comando mesmo, simples e eficiente, valeu demais!!!!

^ | v · Reply · Share ›

Willian Lhorente ➔ **TiagoCeo** · a year ago

Particularmente, gosto bastante do TortoiseGit para Windows, no trabalho uso ele, pois agiliza o trabalho, principalmente quando quero adicionar/commitar somente alguns arquivos.

No PC de casa, uso por linha de comando para aprender trabalhar sem o tortoise, caso precise.

^ | v · Reply · Share ›

Jhonata Matias · a year ago

Melhor guia dentre todos os guias que já vi. Sensacional!

^ | v · Reply · Share ›

David Oliveira · a year ago

que guia incrível! Com certeza essa foi uma das experiências mais inesquecíveis que eu tive com guias! Sensacional, brilhante.

^ | v · Reply · Share ›

Beto da Silva · a year ago

Sensacional. Congrats

^ | v · Reply · Share ›

Valdecir Fernandes Marcia · a year ago

Simples, porém muito eficiente! Valeu!




^ | v · Reply · Share ›

Maurício Eugênio · a year ago

gostoso

^ | v · Reply · Share ›

Load more comments

 [Subscribe](#)  [Add Disqus to your site](#)[Add Disqus](#)  [Privacy](#)