

# GIT

Controle de Versão  
Source Code Management (SCM)



## Apresentação

- Mestre em Ciência da Computação
- Sócio proprietário da Camtwo Sistemas ([camtwo.com.br](http://camtwo.com.br))
- Professor na Universidade do Oeste de Santa Catarina - Unoesc ([unoesc.edu.br](http://unoesc.edu.br))
- A mais de 10 anos trabalhando com Java e web



## Agenda

- Conceitos
- Comandos
- Exemplos
- Exercícios
- Estratégias
- Ferramentas
- Servidores Local e Remoto

## Controles de Versão



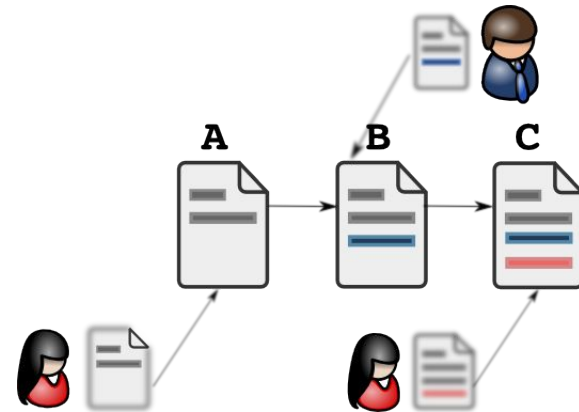
## Origem do GIT

Linus Torvalds (linux)

2005



## Controle de Versão



<http://stoa.usp.br>

## Repositório

Conjunto de arquivos e de todo o histórico de modificações de um projeto

Tem a capacidade de "lembrar" de todas as alterações gravadas nele



## Commit ou Revisão

Cada uma das alterações feitas num arquivo ou repositório

Checkpoint do projeto

Sempre que necessário podemos retroceder a um commit



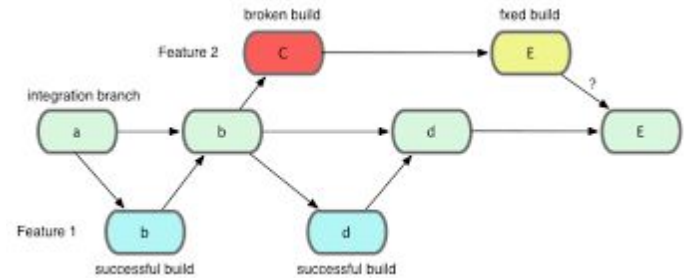
## Cópia de trabalho

Uma cópia do repositório onde são feitas as modificações



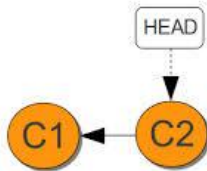
## Branch

É uma ramificação do repositório que pode evoluir de maneira independente



## Head

É uma referência para o commit atual (última revisão)



## Origin

Nome dado ao repositório remoto "primário", onde devem ser feitas as sincronizações.

Nome padrão dados pelo Git

## Fork

É uma bifurcação, uma nova cópia de um projeto para seguir uma nova direção



## Merge

É a capacidade de incorporar alterações do git  
Junção de Branchs



The screenshot shows the Git website homepage. At the top, it says "git -distributed-is-the-new-centralized" with a search bar. Below this, it describes Git as a "free and open source distributed version control system" and lists its features: "easy to learn", "tiny footprint", "lightning fast performance", "outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase", "features like cheap local branching", "convenient staging areas", and "multiple workflows". There is a "Learn Git in your browser for free with Try Git." button. The page is divided into sections: "About" (advantages of Git), "Documentation" (command reference, Pro Git book, videos), "Downloads" (GUI clients, binary releases), and "Community" (bug reporting, mailing list, chat). A "Latest source Release 2.7.1" section is also present. At the bottom, it lists "Companies & Projects Using Git" including Google, Facebook, Microsoft, Twitter, LinkedIn, Netflix, and others.

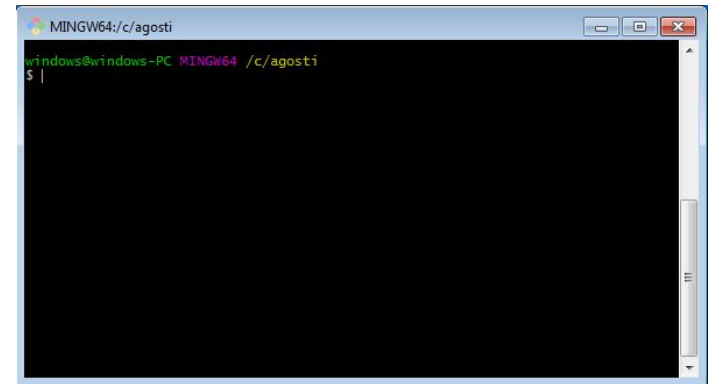
## Vantagens do Git

- Trabalho offline
- Velocidade, menor latência
- Autonomia, histórico fica local
- Facilidade de Mesclagem
- Confiabilidade
- Redução de custos com servidor
- Um mmit de uma alteração dificilmente é perdido
- Tem Integridade checksum (hash sha-1)

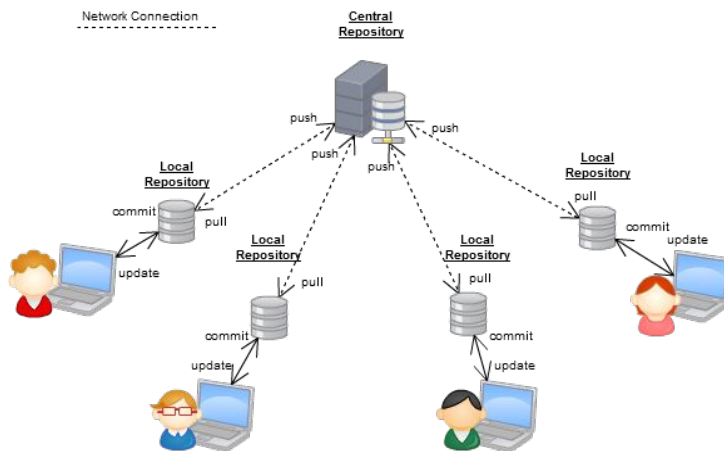
## Devantagens do Git

- Necessidade de maior conhecimento da ferramenta

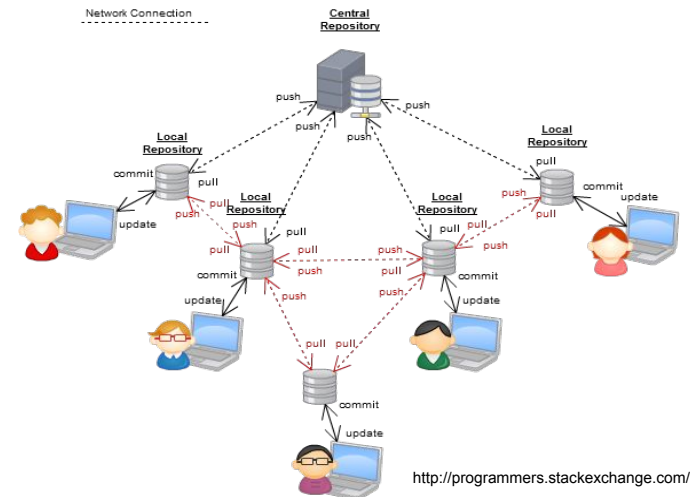
## Git bash



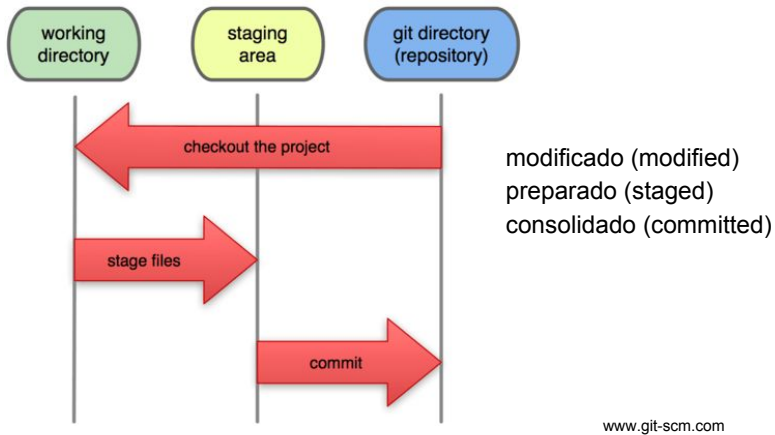
## Repositório e Estação de Trabalho



## Controle de Versão Distribuído



## Três Estados



## Configuração Inicial

```
git config -- global user.name "Cristiano Agosti"
```

```
git config -- global user.email "cristiano.agosti@unoesc.edu.br"
```

```
git config --global color.ui true
```

```
~/.gitconfig
```

## git init

Cria um novo repositório local com um nome especificado

```
git init [nome-do-projeto]
```

```
-- bare
```

```
MINGW64/c/agosti
windows@windows-PC MINGW64 /c/agosti
$ git init
Initialized empty Git repository in C:/agosti/.git/
windows@windows-PC MINGW64 /c/agosti (master)
$ |
```

## git status

Lista todos os arquivos novos ou modificados para serem enviados

```
git status
```

```
MINGW64/c/agosti
windows@windows-PC MINGW64 /c/agosti (master)
$ git status
On branch master

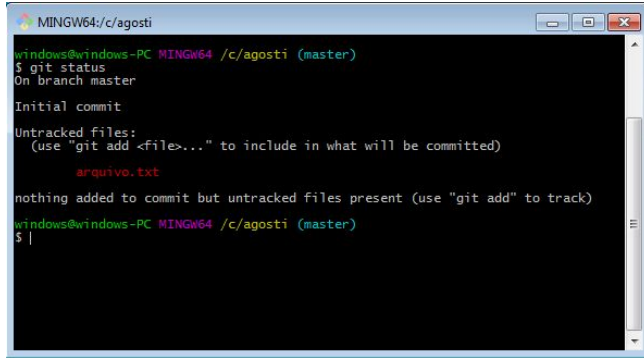
Initial commit

nothing to commit (create/copy files and use "git add" to track)
windows@windows-PC MINGW64 /c/agosti (master)
$ |
```

## git add

Faz o snapshot de um arquivo na preparação para versionamento

git add .



```
MINGW64/c/agosti
windows@windows-PC MINGW64 /c/agosti (master)
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        arquivo.txt

nothing added to commit but untracked files present (use "git add" to track)
$ |
```

## git add



```
MINGW64/c/agosti
windows@windows-PC MINGW64 /c/agosti (master)
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        arquivo.txt

nothing added to commit but untracked files present (use "git add" to track)
windows@windows-PC MINGW64 /c/agosti (master)
$ git add arquivo.txt
windows@windows-PC MINGW64 /c/agosti (master)
$ git status
On branch master

Initial commit

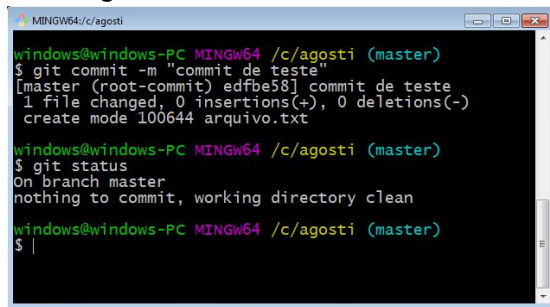
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   arquivo.txt

windows@windows-PC MINGW64 /c/agosti (master)
$
```

## git commit

Grava o snapshot permanentemente do arquivo no histórico de versão

git commit -m "mensagem"



```
MINGW64/c/agosti
windows@windows-PC MINGW64 /c/agosti (master)
$ git commit -m "commit de teste"
[master (root-commit) edf8e58] commit de teste
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 arquivo.txt

windows@windows-PC MINGW64 /c/agosti (master)
$ git status
On branch master
nothing to commit, working directory clean

windows@windows-PC MINGW64 /c/agosti (master)
$ |
```

## git commit --amend

Adiciona mudanças ao último commit

## Prática 1

- a) Criar um repositório chamado pratica\_sobrenome
- b) Verificar o status atual
- c) Criar um arquivo texto chamado descricao\_sobrenome.txt
- d) Verificar o status atual
- e) Enviar para a área de transferência (stage)
- f) Verificar o status atual
- g) Enviar para o repositório

## .gitignore

Ignora arquivos e diretórios temporários

Um arquivo de texto chamado .gitignore suprime o versionamento acidental de arquivos e diretórios correspondentes aos padrões especificados

Criado na raiz do repositório

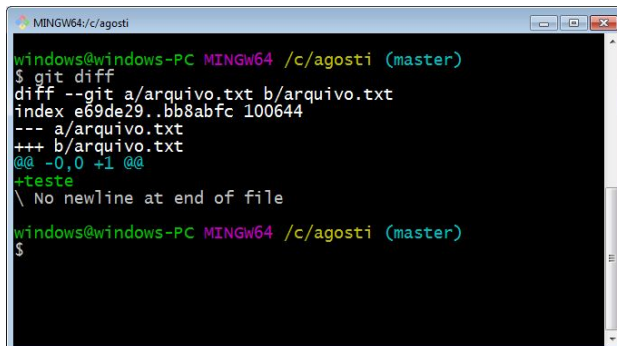
```
*.log  
build/  
temp-*
```

Lista Arquivos ignorados

git ls-files --other --ignored --exclude-standard

## git diff

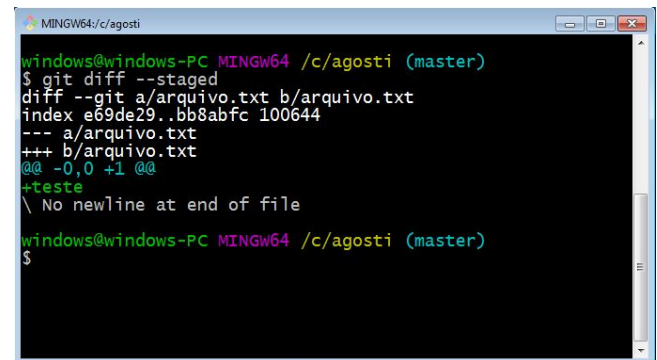
Mostra mudanças realizadas em arquivos e que ainda não foram versionadas



```
mingw64/c/agosti  
windows@windows-PC MINGW64 /c/agosti (master)  
$ git diff  
diff --git a/arquivo.txt b/arquivo.txt  
index e69de29..bb8abfc 100644  
--- a/arquivo.txt  
+++ b/arquivo.txt  
@@ -0,0 +1 @@  
+teste  
\ No newline at end of file  
windows@windows-PC MINGW64 /c/agosti (master)  
$
```

## git diff --staged

Mostra a diferença entre arquivos preparados e suas últimas versões




```
mingw64/c/agosti  
windows@windows-PC MINGW64 /c/agosti (master)  
$ git diff --staged  
diff --git a/arquivo.txt b/arquivo.txt  
index e69de29..bb8abfc 100644  
--- a/arquivo.txt  
+++ b/arquivo.txt  
@@ -0,0 +1 @@  
+teste  
\ No newline at end of file  
windows@windows-PC MINGW64 /c/agosti (master)  
$
```



## git log

Lista o histórico de versões para o branch atual



```
mingw64/c/agosti
windows@windows-PC MINGW64 /c/agosti (master)
$ git log
commit edf8e5882b2444a1cde95ae6a3e2c39bc0d36713
Author: cristiano agosti <cristiano.agosti@gmail.com>
Date: Mon Feb 8 21:44:21 2016 -0200

    commit de teste

windows@windows-PC MINGW64 /c/agosti (master)
$
```

## git log

git log -p comando log com diff

git log -p -2 restringe número de commits

git log --follow [arquivo] lista o histórico de versões para um arquivo, incluindo mudanças de nome

git diff [primeiro-branch]...[segundo-branch] mostra a diferença de conteúdo entre dois branches

git show [commit] retorna mudanças de metadata e conteúdo para o commit especificado

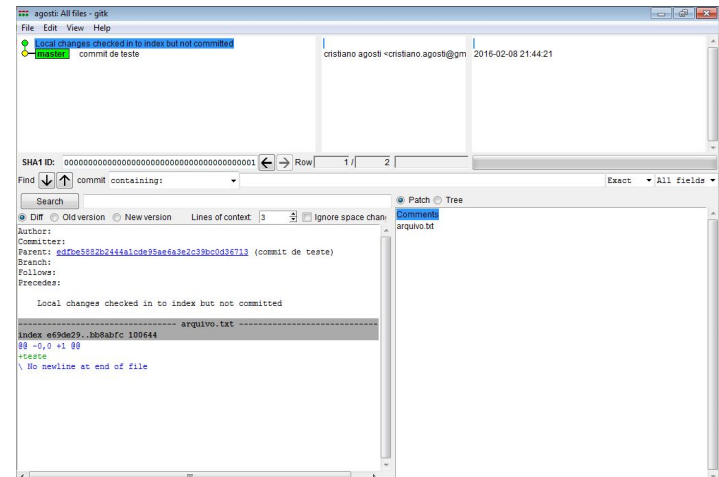
git log --pretty=oneline mostra em uma linha

## git blame

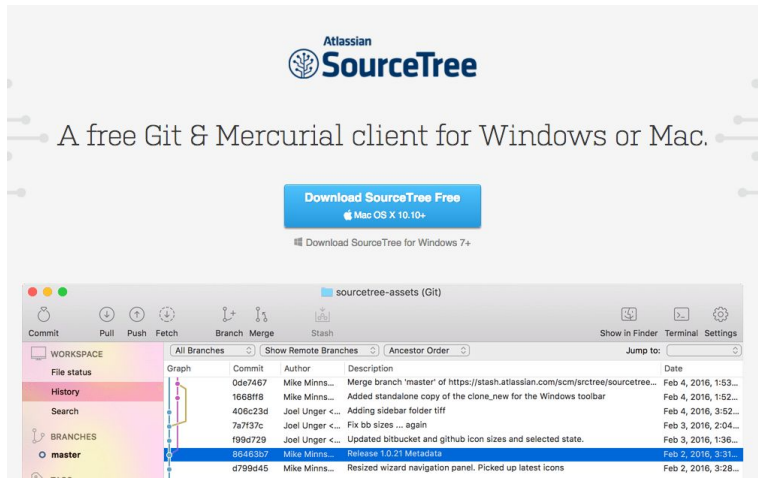
Listagem de modificações de um arquivo

git blame nome-do-arquivo

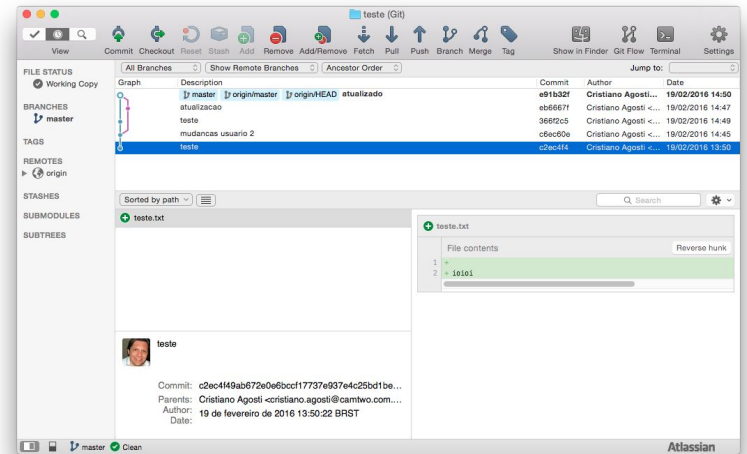
## gitk



## Source Tree (<https://www.sourcetreeapp.com>)



## Source Tree



## git reset

Retira o arquivo da área de preparação, mas preserva seu conteúdo

git reset [arquivo]

git reset HEAD [arquivo]

git reset --hard [commit]

## git checkout

Volta a versão do arquivo anterior

git checkout -- [nome\_do\_arquivo]

## git rm

`git rm [arquivo]`

Remove o arquivo do diretório de trabalho e o prepara a remoção

`git rm --cached [arquivo]`

Remove o arquivo do controle de versão mas preserva o arquivo localmente

`git mv [arquivo-original] [arquivo-renomeado]`

Muda o nome do arquivo e o prepara para o commit

## Prática 2

Sobre o repositório criado na prática 1 faça:

- Criar novo arquivo chamado `renomear_sobrenome.txt`, enviar para o repositório, renomear para `renomeado_sobrenome.txt`, enviar para o repositório
- Criar novo arquivo chamado `alterar1.sobrenome.txt`, enviar para o repositório, alterar o conteúdo, voltar a alteração
- Criar novo arquivo chamado `alterar2_sobrenome.txt`, enviar para o repositório, alterar o conteúdo e enviar para a área de stage, voltar o conteúdo anterior, alterar novamente e enviar para o repositório, voltar o conteúdo enviado ao repositório
- Criar um arquivo compactado com extensão zip, criar arquivo `gitignore` ignorando esse arquivo, enviar para o repositório
- Criar novo arquivo chamado `excluir_sobrenome.txt`, enviar para o repositório, excluir do repositório, enviar para o repositório

## git tag

criação de pontos de marcação (ponteiros) para releases

`git tag`

`git tag -a v1.0 -m "versao 1.0"` (cria tag no commit atual)

`git show v1.0`

`git tag -a v1.2 9fceb02` (commit antigo)

`git checkout v1.0`

`git tag -d v1.0`

## git branch / merge / checkout

cria uma nova ramificação para o código atual

**git branch**

Lista todos os branches locais no repositório atual

**git branch** [nome-do-branch] ou **git checkout -b** [nome-do-branch]

Cria um novo branch

**git checkout** [nome-do-branch]

Muda para o branch especificado e atualiza o diretório de trabalho

**git merge** [nome-do-branch]

Combina o histórico do branch especificado ao branch atual

**git branch -d** [nome-do-branch]

Exclui o branch especificado

## Resolução de Conflito

Ocorre quando alterações são realizadas na mesma parte de código nesse caso devemos atuar manualmente para resolve-lo

```

windows@windows-PC MINGW64 /c/agosti (master)
$ git merge -m "merge com teste" teste
Auto-merging arquivo.txt
CONFLICT (content): Merge conflict in arquivo.txt
Automatic merge failed; fix conflicts and then commit the result.

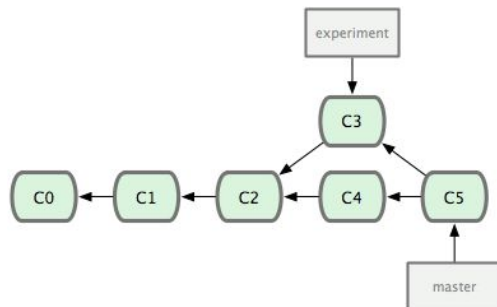
windows@windows-PC MINGW64 /c/agosti (master|MERGING)
$
```

## Resolução de Conflito

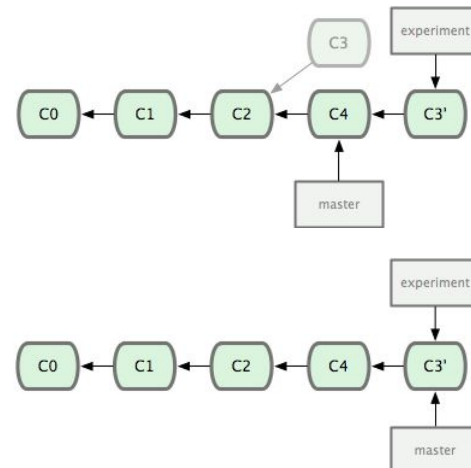
```

<<<<<< HEAD
function master() {
    return 0;
}
=====
function teste() {
}
>>>>>> teste
```

## Merge



## Rebase



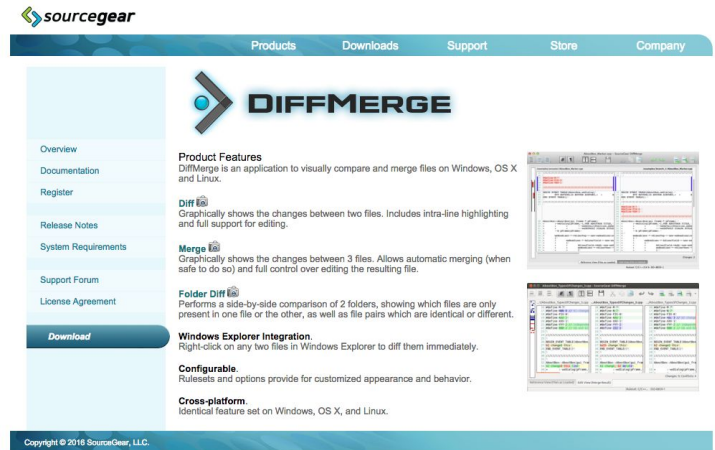
## Rebase

**Não faça rebase de commits que você enviou para um repositório público.**

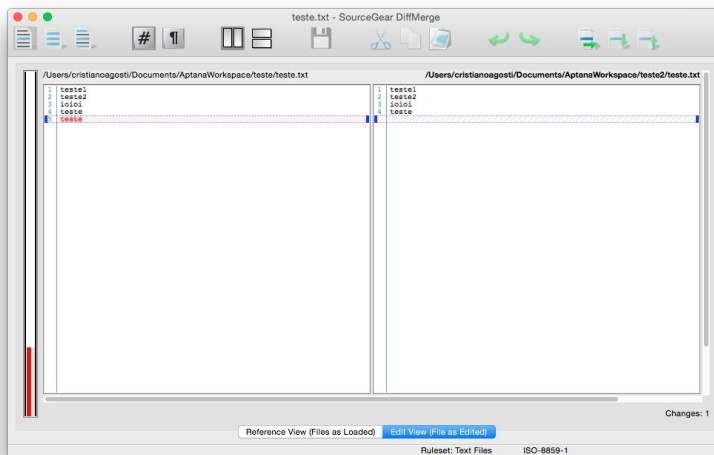
Se você seguir essa regra você ficará bem. Se não seguir, as pessoas te odiarão e você será desprezado por amigos e familiares.

git-scm.com

## Diff Merge (<https://sourcegear.com/diffmerge>)



## Diff Merge



## Mostrando o branch no prompt Linux e Os-x

.bash\_profile

# Git branch in prompt.

```
parse_git_branch() {  
    git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\) / (\1)/'  
}
```

### Prática 3

- a) Criar uma tag para a versão 1.0
- b) Criar uma nova branch
- c) Ir para a nova branch
- d) Criar um novo arquivo para a branch
- e) Enviar para o repositório
- f) Ir para o master
- g) Fazer mudança no mesmo arquivo
- h) Enviar mudanças
- i) Fazer o merge com o master
- j) Resolver conflitos

### Git em Servidor Local

**git init** --bare inicializa o repositório central preparando para receber mudanças

**git clone file:///Volumes/teste/projeto.git/ nome-projeto** cria uma cópia local

**git clone** projeto-git projeto-git-destino

**git remote** identifica o nome (alias) do servidor remoto

### stash

Comando que possibilita ajuda no momento em que desejamos mudar de branch sem perder o que estamos fazendo e sem dar o commit

Forma de backup de um código ainda não finalizado e que ainda não queremos realizar o commit

### stash

#### Criando

git stash

#### Listando

git stash list

stash@{0}: WIP on master: 049d078 added the index file

stash@{1}: WIP on master: c264051... Revert "added file\_size"

stash@{2}: WIP on master: 21d80a5... added number to log

#### Aplicando

git stash apply

git stash apply stash@{2}

git stash apply --index

## stash

### Excluindo

```
git stash drop stash@{0}  
git stash pop
```

### Revertendo

```
git stash show -p stash@{0} | git apply -R  
git stash show -p | git apply -R
```

Criando um branch: `git stash branch testchanges`

## alias

`<repo>/ .git/config` – Repository-specific settings.

`~/.gitconfig` – User-specific settings. This is where options set with the `--global` flag are stored.

`$(prefix)/etc/gitconfig` – System-wide settings.

### [alias]

```
st = status  
co = checkout  
br = branch  
up = rebase  
ci = commit
```

```
git config --global alias.stash-unapply '!git stash show -p | git apply -R'  
git stash-unapply
```

## Git em Servidor Local

**git push** [alias remoto] [branch] envia as mudanças para o servidor remoto

**git pull** [alias remoto] [branch] busca mudanças

**git fetch** [alias remoto] [branch] busca mudanças sem commit

## Prática 4

- Configurar e criar o servidor remoto
- Clonar simulando o usuário 1, realizar mudanças, enviar para o servidor
- Clonar simulando o usuário 2, realizar mudanças, enviar para o servidor
- Simular e gerar conflitos e solucionar os conflitos entre os usuários

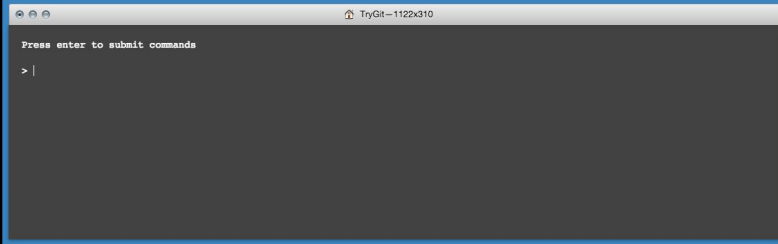
<https://try.github.io>

## 1.1 · Got 15 minutes and want to learn Git?

Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes. It's a distributed version control system.

Our terminal prompt below is currently in a directory we decided to name "octobox". To initialize a Git repository here, type the following command:

`git init`



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

## 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, nederlandse, polski, pyccckii, türkçe,

日本語, 中文, 한국어

por favor informe problemas em [github](#)



download the  
cheat sheet  
now. it's free!

want a simple  
but powerful  
git client for  
your mac?



**Are You a Front-End Developer?**  
by Roger Dudler, Author of the Git Simple Guide

Try Frontify

Now Free with  
GitHub Integration!



<http://aprenda.vidageek.net/aprenda/git>

Aprenda Git Login Sobre

Avise! Estamos em Beta. Caso encontre algum problema, envie um email para [games@vidageek.net](mailto:games@vidageek.net)

## Aprenda Git

Git é uma ferramenta de controle de versão que tem crescido muito nos últimos anos. Este jogo cobre os principais comandos e fluxos de trabalho com esta ferramenta.

Você pode começar os exercícios pelo [primeiro](#) ou escolher um grupo abaixo:

### Conteúdo:

- [Criar um repositório](#)
- [Adicionar arquivos](#)
- [Fazer Commits](#)
- [Criar Branches](#)
- [Mudar de branch ativo](#)
- [Merge de branches](#)
- [Rebase de branches](#)
- [Enviar commits para branches remotos \(Push\)](#)
- [Pegar commits de branches remotos \(Pull\)](#)
- [Git Workflow](#)

[https://training.github.com/kit/downloads/pt\\_BR/github-git-cheat-sheet](https://training.github.com/kit/downloads/pt_BR/github-git-cheat-sheet)

GitHub Training

Trainers Classes Schedule Kit Contact

## Folha de Dicas de Git do GitHub (pt-BR)

Git é um sistema de controle de versão distribuído open source que facilita ações com o GitHub em seu notebook ou desktop. Esta folha de dicas resume instruções comumente usadas via linha de comando do Git para referência rápida.

### Instale o Git

GitHub fornece clientes desktop que incluem uma interface gráfica para as ações mais comuns em um repositório e atualiza automaticamente para a linha de comando do Git para cenários avançados.

### GitHub para Windows

<http://windows.github.com>

### GitHub para Mac

<http://mac.github.com>

Distribuições do Git para Linux e sistemas POSIX são disponíveis no site oficial do Git SCM.

### Git para todas as plataformas

<http://git-scm.com>

### Configure a ferramenta

Configure informações de usuário para todos os repositórios locais

```
$ git config --global user.name "[nome]"
```

### Faça mudanças

Revise edições e crie uma transação de commit

```
$ git status
```

Lista todos os arquivos novos ou modificados para serem commitados

```
$ git diff
```

Mostra diferenças no arquivo que ainda não foram preparadas

```
$ git add [arquivo]
```

Faz o snapshot de um arquivo na preparação para versionamento

```
$ git diff --staged
```

Mostra a diferença entre arquivos preparados e suas últimas versões

```
$ git reset [arquivo]
```

Retira o arquivo da área de preparação, mas preserva seu conteúdo

```
$ git commit -m "[mensagem descritiva]"
```

Grava o snapshot permanentemente do arquivo no histórico de versão

### Mudanças em grupo

Nomeie uma série de commits e combine os esforços completos



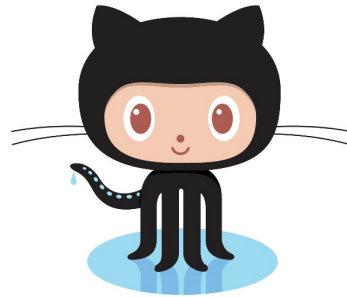
## Github

Rede Social para desenvolvedores

[github.com](https://github.com)

Criar conta

Chave Acesso



## ssh-keygen

```
MINGW64:/c/agosti
windows@windows-PC MINGW64 /c/agosti (master|MERGING)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/windows/.ssh/
id_rsa):
Created directory '/c/Users/windows/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/windows/.ssh
/id_rsa.
Your public key has been saved in /c/Users/windows/.ssh/id_
rsa.pub.
The key fingerprint is:
SHA256:Lyd0LTdsQ9c9X5ivcle8pUka9g7k5Spr4ldskRsyysQ windows@
windows-PC
The key's randomart image is:
+--[RSA 2048]-----
  .o+=.
  .+ +.
  .+ =+.
  E . . . o+
  +.S.o.B . o
  =+00+ +.
  +00+ +.
  o .o+ o .
+--[SHA256]-----
windows@windows-PC MINGW64 /c/agosti (master|MERGING)
$ |
```

## id\_rsa.pub

ssh-rsa

```
AAAAB3NzaC1yc2DAQAABAAABAQDE/OqUCyeXE4ZVZ7nO
TdYJlgeO3xzNPOEKPUi9NGzHZ9R9T5aYD2jFUewPdo+Pz
oEoQDvLK6jg7jyShmiD+OQ2717lyK6cDac3OkcFdIp8bMHL
Audxxf7sjhFt+WWIL6vmJS4Jpe9DOIj7xPHx+KzU/WBHCclx
Ves8IBUtJkkkg6/t1UMtAOomPWcRDULiIRFqt71mdf3c4IUsEE
Y6tTnZZwut9LHDUbq/K2gV3OcdcIJR6j79mNcVLxQjBB9o30
SNzCn6Xvk3bCHDEO/pkmZIRTFjZzeXNXCLn4meXwJlZq/b0
2ztCTaeT9kX7cEZUe1uzTs3wX windows@windows-PC
```

## Configurando a Chave

Personal settings

Profile

Account settings

Emails

Notification center

Billing

SSH keys

Security

Applications

Personal access tokens

Repositories

Organizations

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

agosti

17:e7:e1:e8:68:80:4d:5a:c8:25:d2:57:39:82:4a:2c

Added on Jun 1, 2012 — No recent activity

Delete

agostimac

eb:4a:1e:93:8d:9c:cf:7b:54:45:9c:bb:c1:5b:07:5a

Added on Jul 6, 2013 — No recent activity

Delete

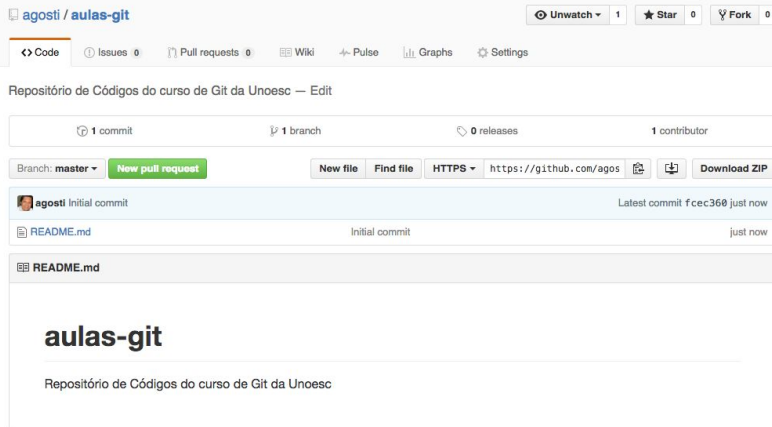
Title

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key

## Repositório Github Criado



## Clonando Repositório do Github (usando ssh)

```
git clone git@github.com:agosti/aulas-git.git
```

## Prática 5

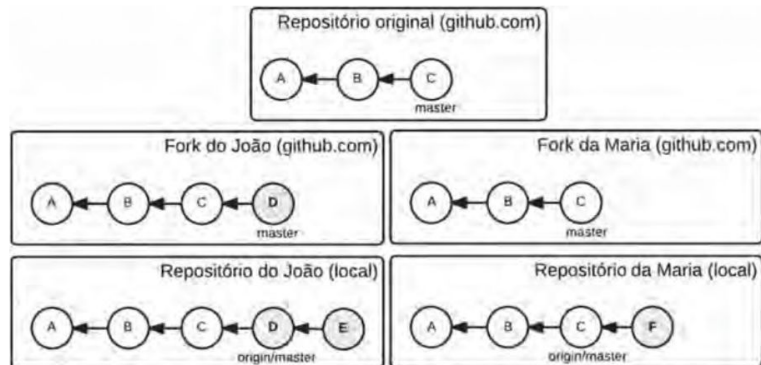
- criar conta no Github
- criar repositório
- clonar
- modificar local
- enviar
- clonar projeto do git do Professor, criar novo arquivo sobrenome.txt, enviar para o servidor
- simular conflitos entre colegas

## Colaborar com Projetos Open Source

**Fork** clonar para outro repositório

**Pull Request** enviar as modificações

## Fork e Pull Request

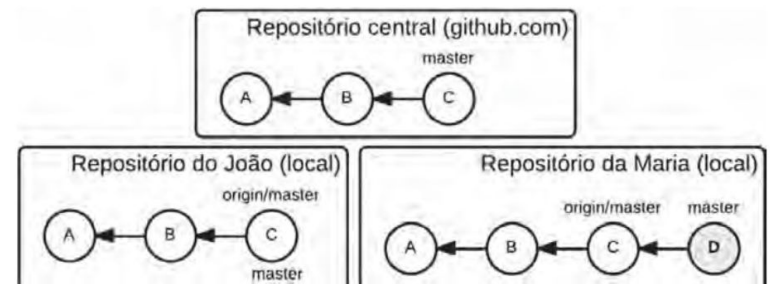


## Prática 6

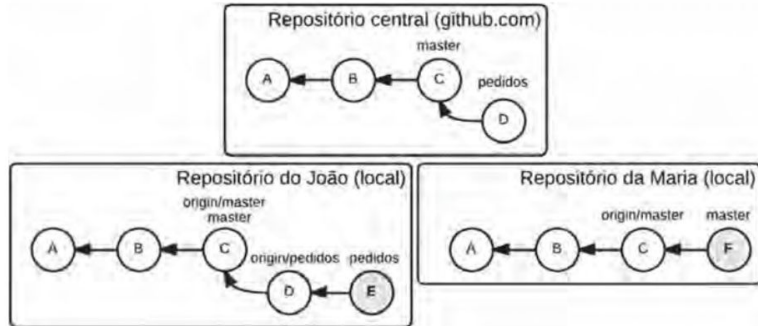
- Criar o Fork de um projeto
- Modificar
- Enviar Fork
- Realizar o Pull Request

## Estratégias

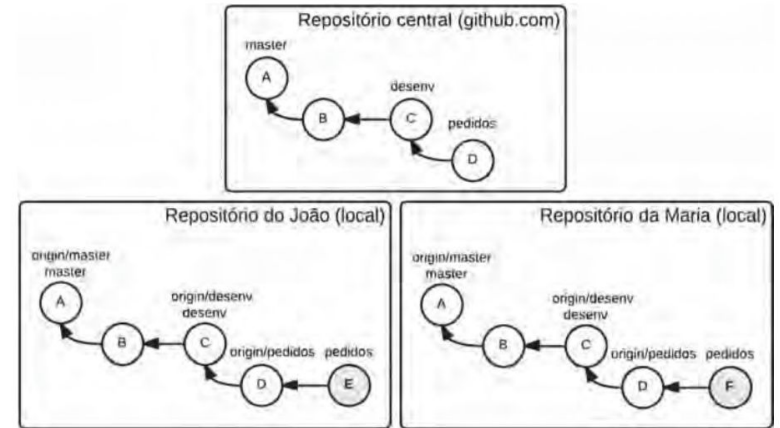
### Utilizando só a branch master com um repositório central



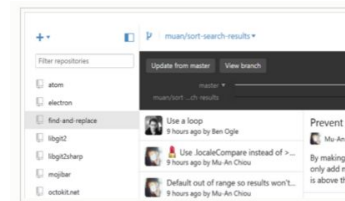
## Utilizando branches por funcionalidade com um repositório central



## Utilizando branches por etapa de desenvolvimento com um repositório central

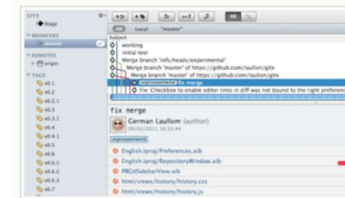


## Ferramentas Gráficas



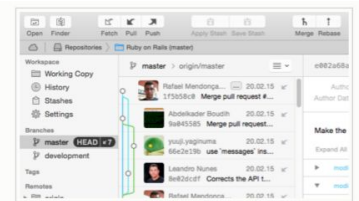
### GitHub Desktop

Platforms: Windows, Mac  
Price: Free



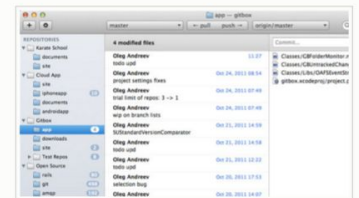
### GitX-dev

Platforms: Mac  
Price: Free



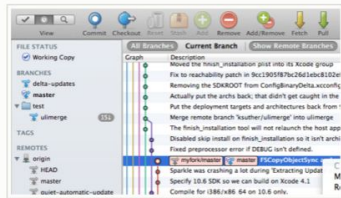
### Tower

Platforms: Mac  
Price: \$69/user (Free 30 day trial)



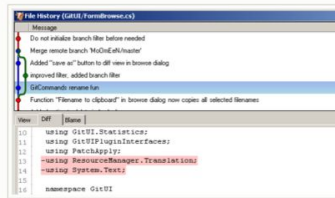
### Gitbox

Platforms: Mac  
Price: \$14.99



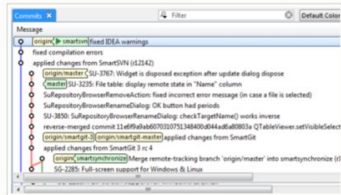
### SourceTree

Platforms: Mac, Windows  
Price: Free



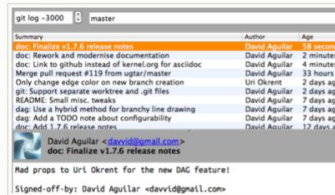
### Git Extensions

Platforms: Windows  
Price: Free



### SmartGit

Platforms: Windows, Mac, Linux  
Price: \$79/user / Free for non-commercial use



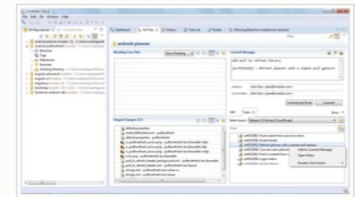
### git-cola

Platforms: Windows, Mac, Linux  
Price: Free



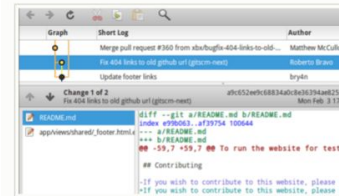
### GitUp

Platforms: Mac  
Price: Free



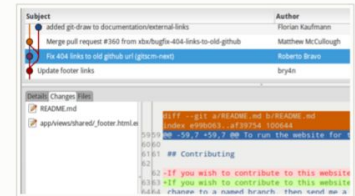
### GitEye

Platforms: Windows, Mac, Linux  
Price: Free



### giggle

Platforms: Linux  
Price: Free



### gitg

Platforms: Linux  
Price: Free

## Protocolos do Git

Local (file)

Git

SSH

Http(S)

## Criando um Daemon

git daemon --base-path=. --export-all --enable=receive-pack  
--reuseaddr --informative-errors --verbose  
[8322] Ready to rumble

git clone git://IP\_DA\_MAUQUINA/projeto.git

<http://railsware.com/blog/2013/09/19/taming-the-git-daemon-to-quickly-share-git-repository>

## Referências

BELL, Peter, BRENT, Beer. Introdução ao GitHub. Um guia que não é técnico. Novatec, 2015. ISBN: 9788575224144

AQUILES, Alexandre, FERREIRA, Rodrigo. Controlando versões com Git e Github. Editora Casa do Código, 2014, ISBN 9788566250534

## Links

- [http://rogerdudler.github.io/git-guide/index.pt\\_BR.html](http://rogerdudler.github.io/git-guide/index.pt_BR.html)
- <https://git-scm.com/book/pt-br/v1>
- <https://try.github.io>
- <http://aprenda.vidageek.net/aprenda/git>
- [https://training.github.com/kit/downloads/pt\\_BR/github-git-cheat-sheet.html](https://training.github.com/kit/downloads/pt_BR/github-git-cheat-sheet.html)
- <http://www.loiane.com/2013/11/screencast-git-e-github-para-iniciantes>

## Contato

- skype: cristiano.agosti
- twitter: @agosti
- cristiano.agosti@unoesc.edu.br
- facebook.com/cristiano.agosti