

GIT

Controle de Versão
Source Code Management (SCM)



Apresentação

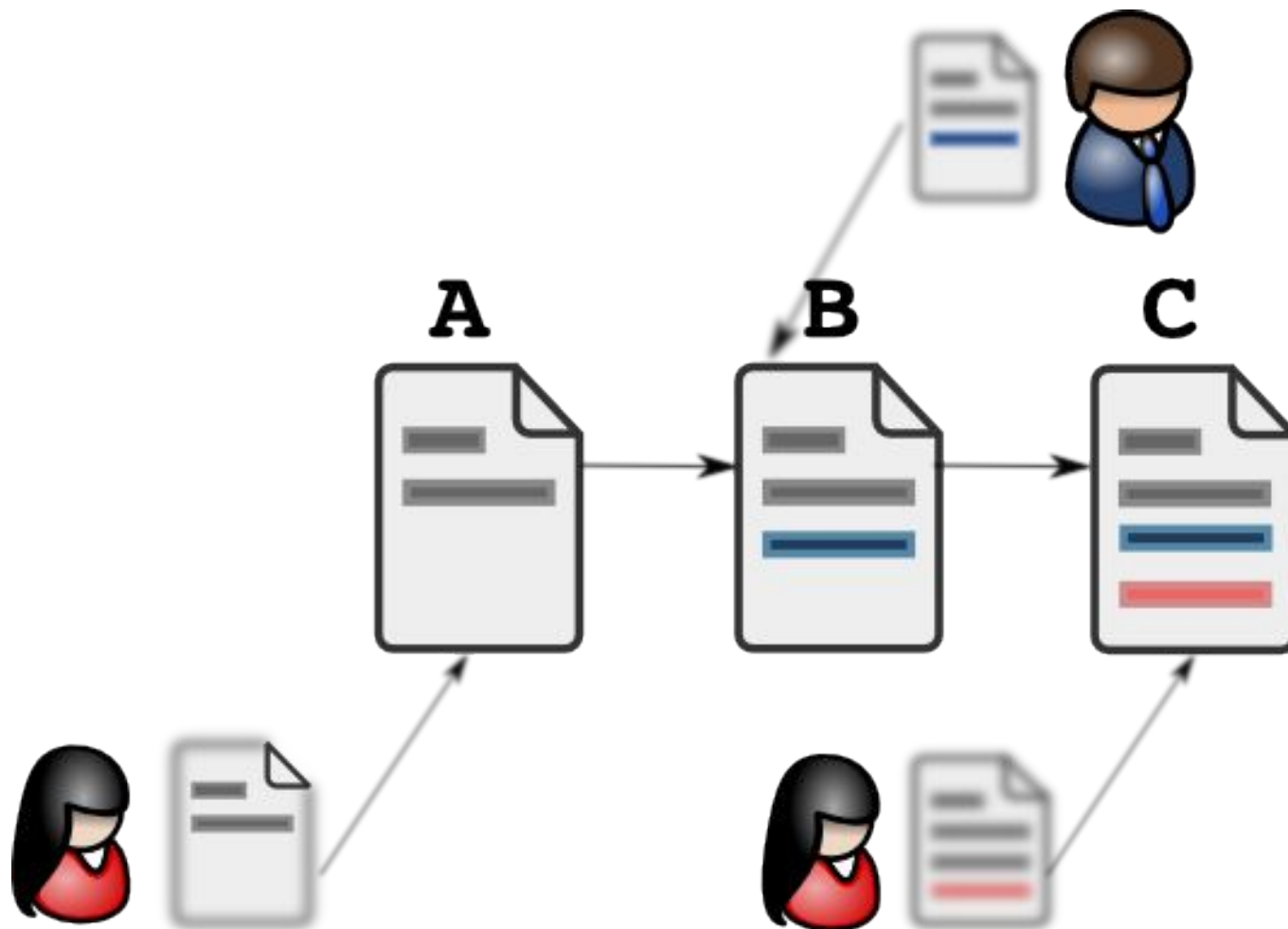
- Mestre em Ciência da Computação
- Sócio proprietário da Camtwo Sistemas (camtwo.com.br)
- Professor na Unoesc (unoesc.edu.br)
- A mais de 10 anos trabalhando com Java e web



Agenda

- Conceitos
- Comandos
- Exemplos
- Exercícios
- Estratégias
- Ferramentas
- Servidores

Controle de Versão



Softwares para Controle de Versão



git

Origem do GIT

Linus Torvalds (linux)

2005



Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



Learn Git in your browser for free with **Try Git**.



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).



Mac GUIs



Tarballs



Windows Build



Source Code

Companies & Projects Using Git

Google

facebook

Microsoft

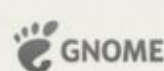
twitter

LinkedIn

NETFLIX



PostgreSQL



Vantagens do Git

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

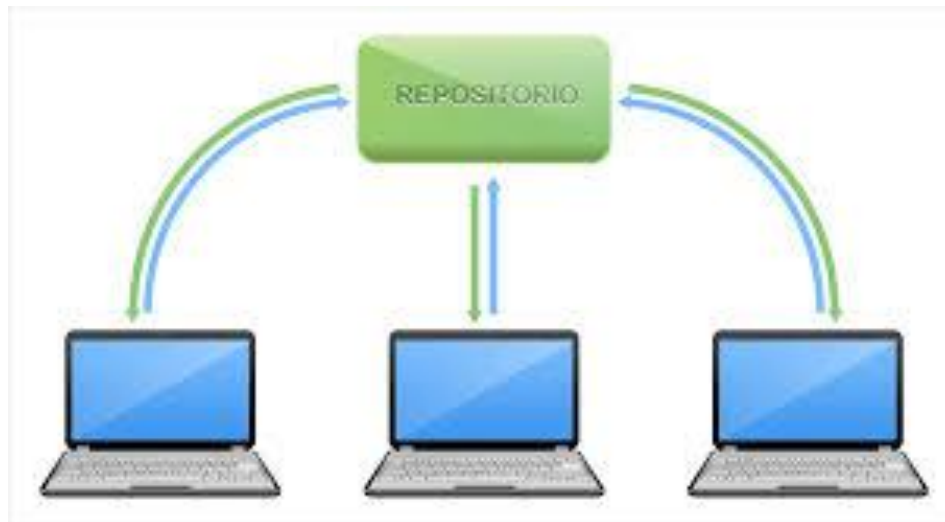
Desvantagens do Git

- Necessidade de maior conhecimento da ferramenta

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

Identificado por hash



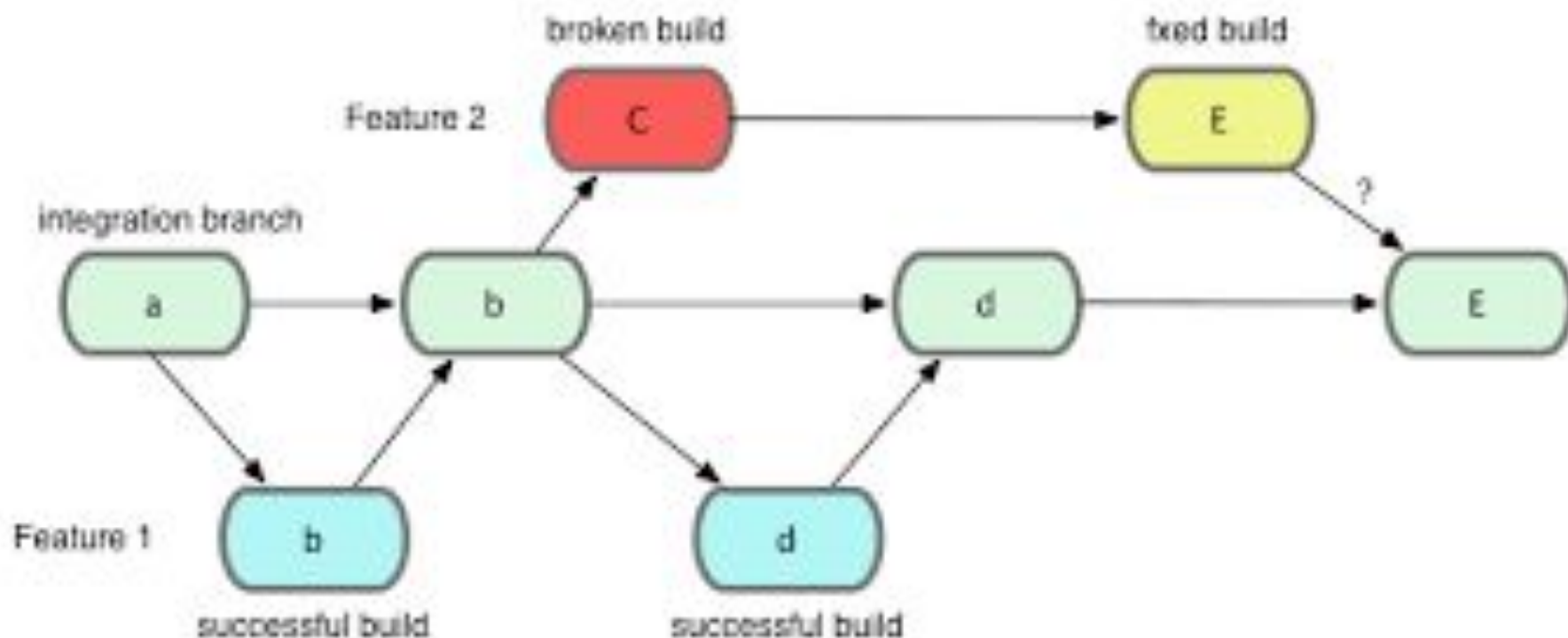
Cópia de trabalho (clone)

Uma cópia do repositório onde são feitas as modificações



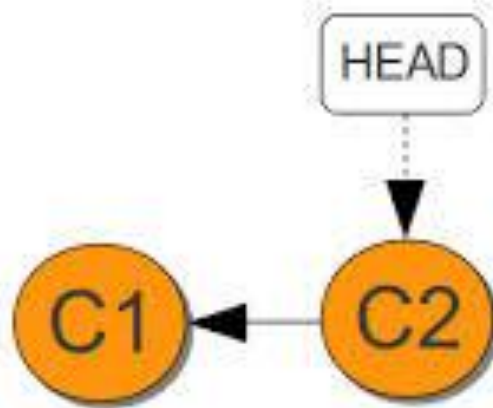
Branch

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 dado 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

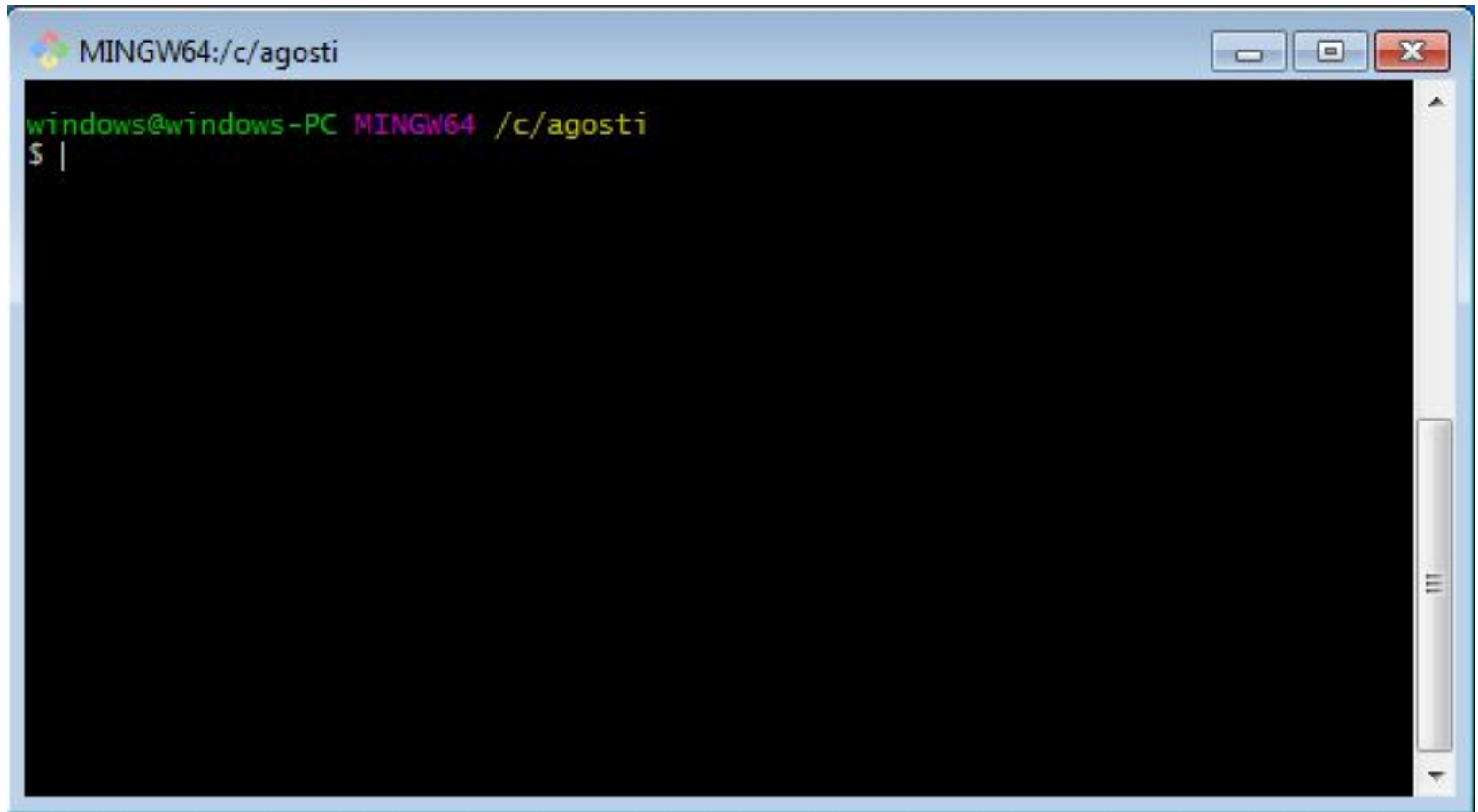


Pull

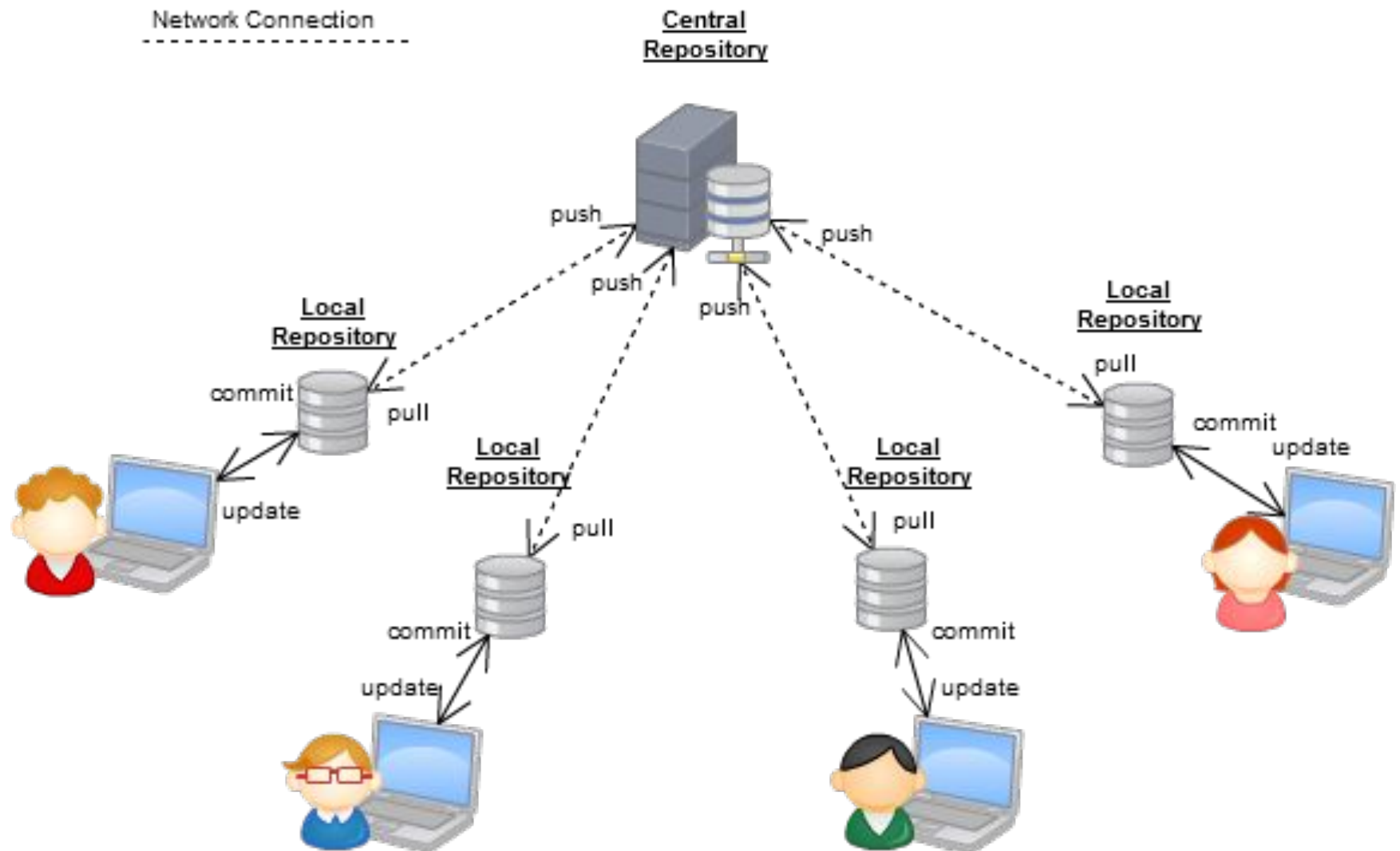
Busca todas as modificações feitas do servidor



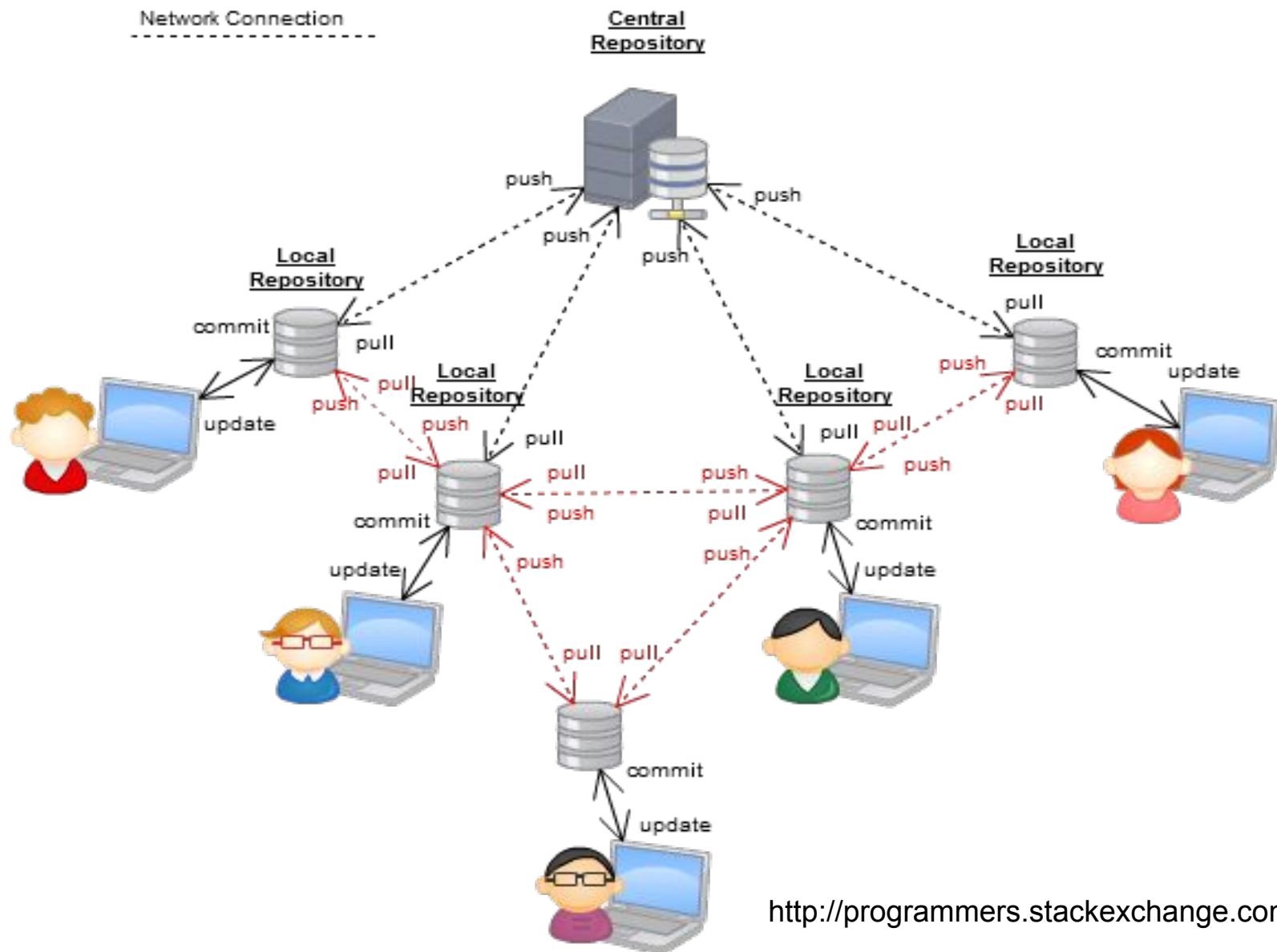
Git bash



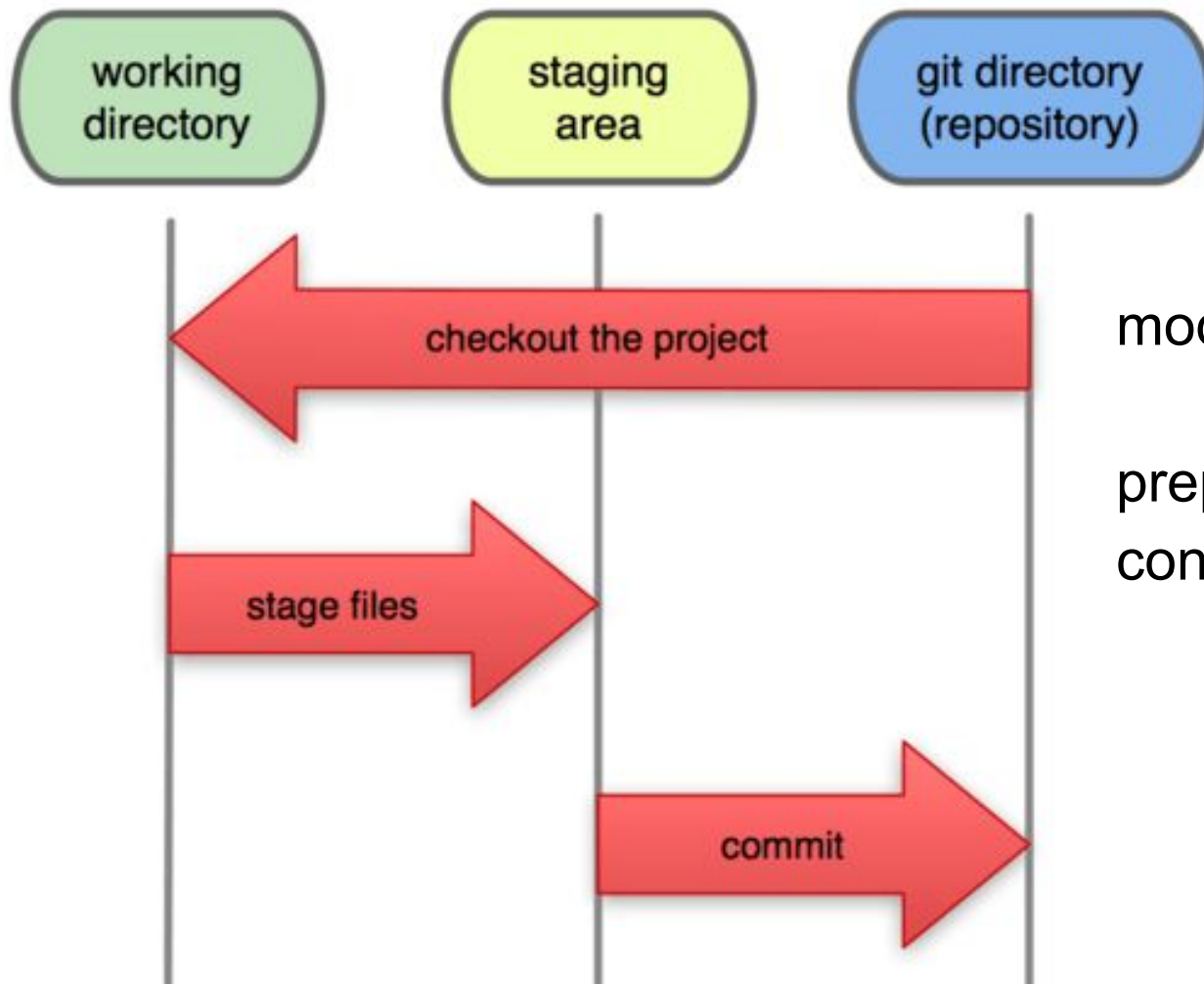
Repositório e Estação de Trabalho



Controle de Versão Distribuído



Três Estados



modificado

(modified - working)

preparado (staged)

consolidado (committed)

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
```

```
git config --global credential.helper cache
```

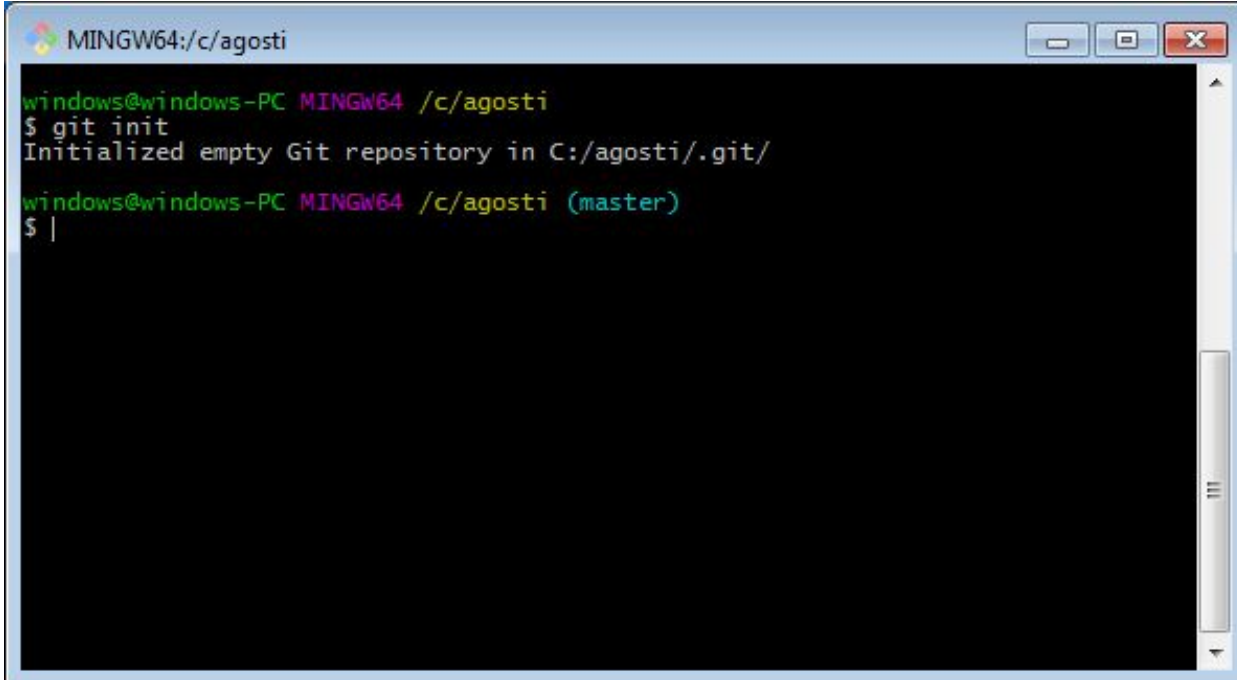
```
~/.gitconfig
```

git init

Cria um novo repositório local com um nome especificado

git init [nome-do-projeto]

-- bare (servidor)

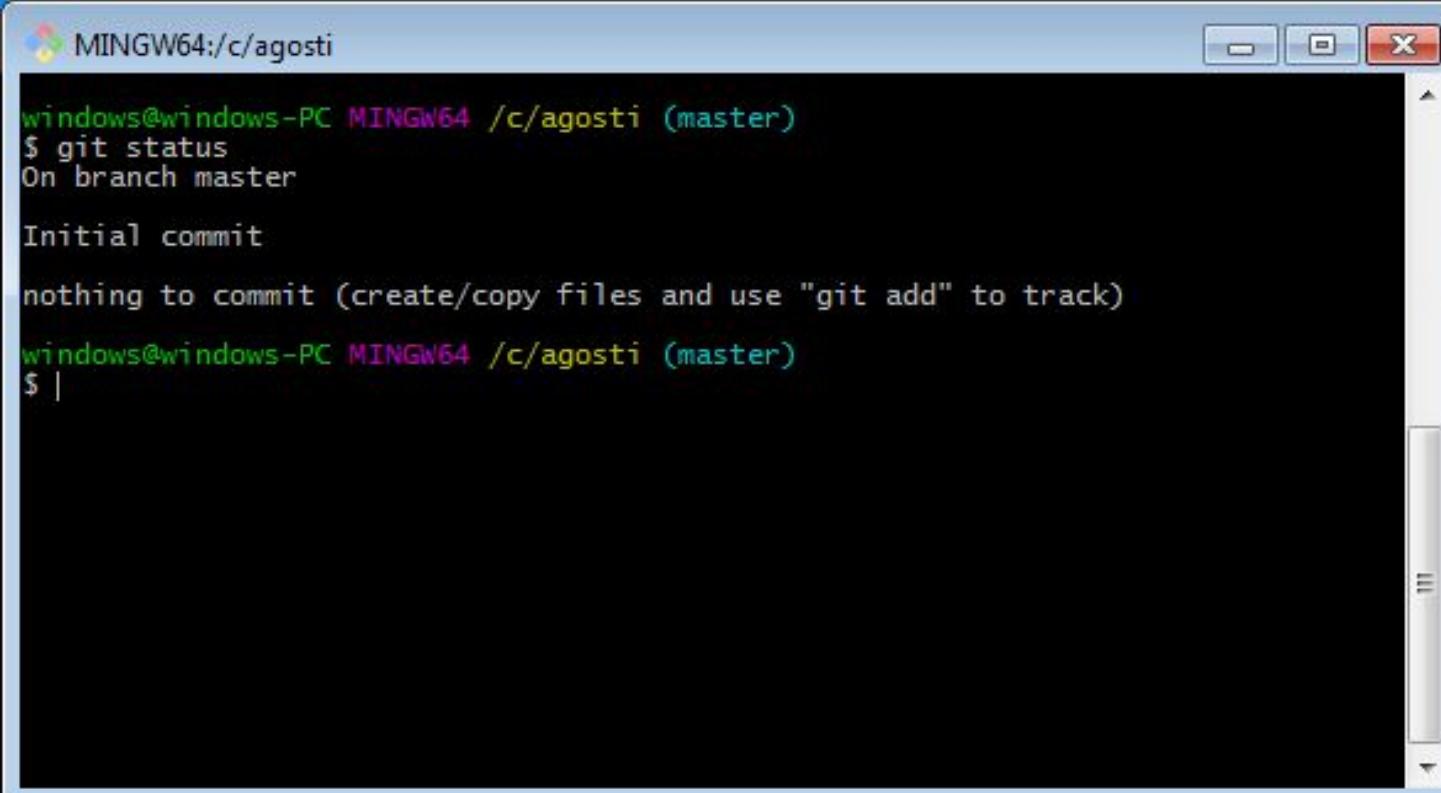
A screenshot of a MINGW64 terminal window titled 'MINGW64:/c/agosti'. The terminal shows the following commands and output:

```
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

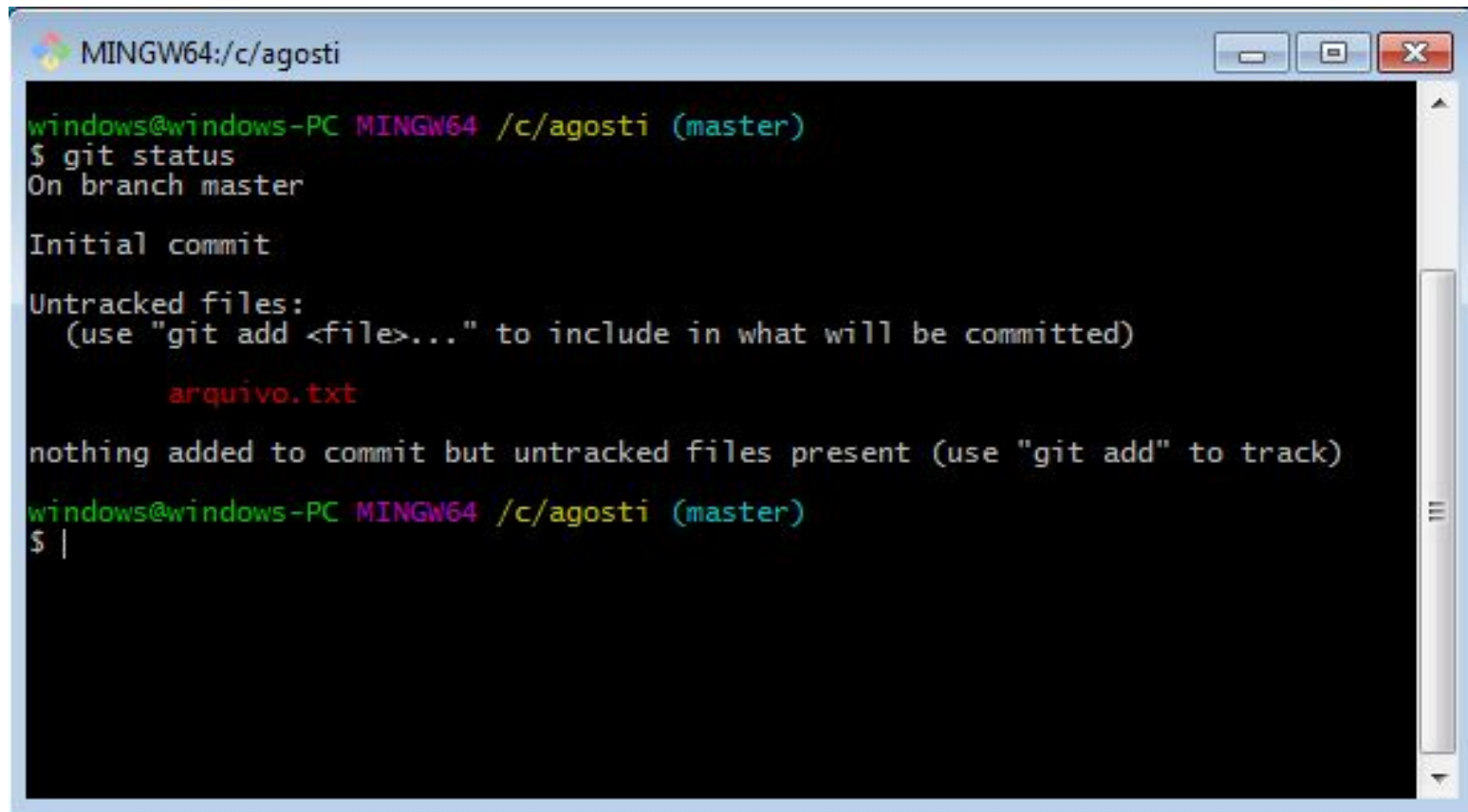
nothing to commit (create/copy files and use "git add" to track)

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

git add

Faz o snapshot de um arquivo na preparação para versionamento

git add .

A screenshot of a MINGW64 terminal window. The title bar shows 'MINGW64:/c/agosti'. The terminal output shows the command 'git status' being executed, resulting in an 'Initial commit' state. It lists 'arquivo.txt' as an untracked file. The message 'nothing added to commit but untracked files present' is displayed, followed by a prompt for the next command.

```
mingw64@windows-PC MINGW64 /c/agosti (master)
$ git status
On branch master

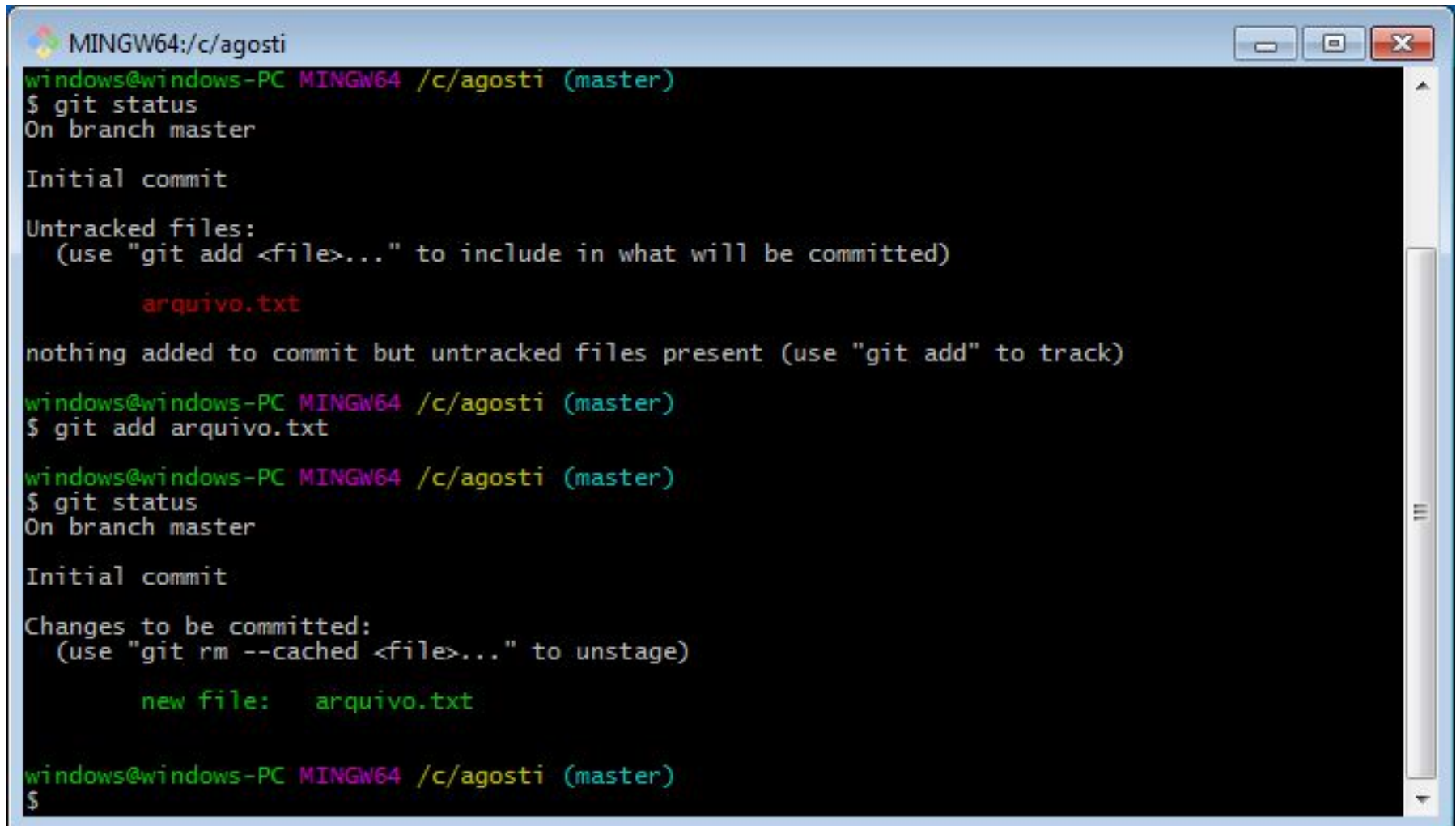
Initial commit

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)
mingw64@windows-PC MINGW64 /c/agosti (master)
$ |
```

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)

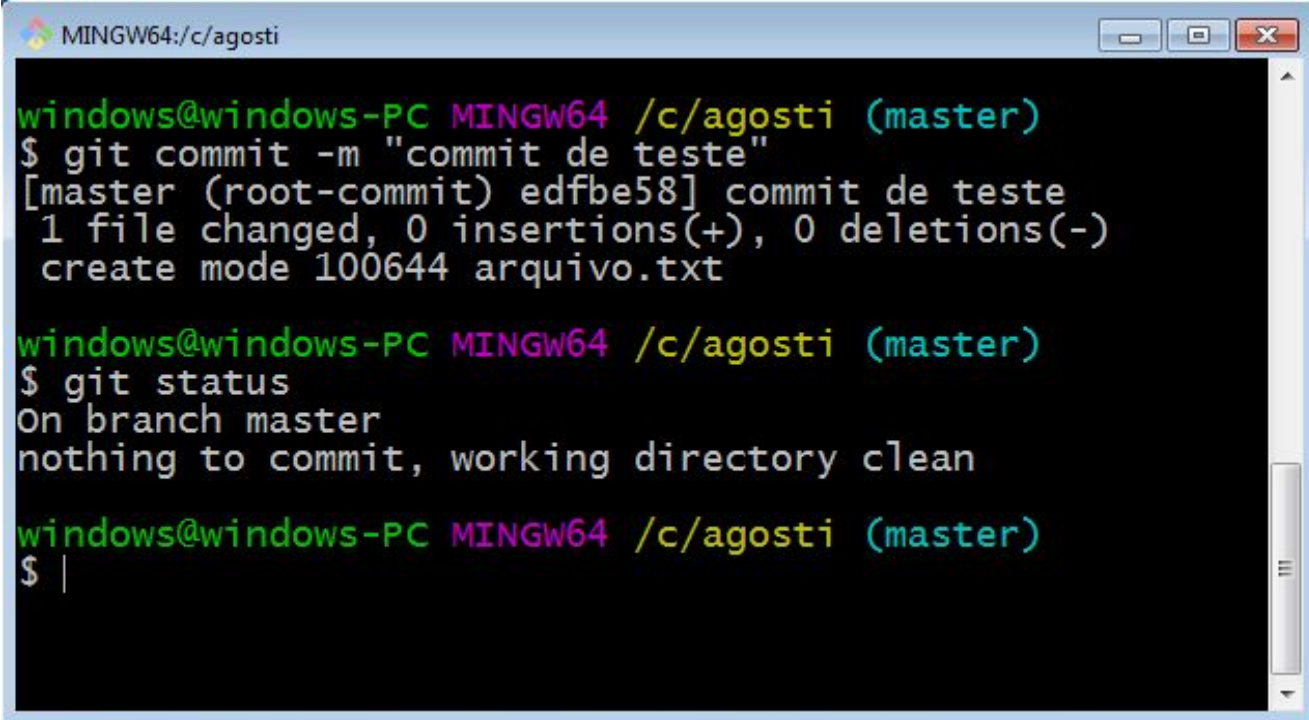
        new file:   arquivo.txt

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

git commit

Grava o snapshot permanentemente do arquivo no histórico de versão

`git commit -m "mensagem"`

A screenshot of a Windows command prompt window titled "MINGW64:/c/agosti". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows a series of commands and their outputs. The first command is "git commit -m 'commit de teste'", which results in a commit message "[master (root-commit) edfbe58] commit de teste" and a summary "1 file changed, 0 insertions(+), 0 deletions(-) create mode 100644 arquivo.txt". The second command is "git status", which outputs "On branch master" and "nothing to commit, working directory clean". The third command is a partial "git" command, showing only "\$ |".

```
mingw64@windows-PC MINGW64 /c/agosti (master)
$ git commit -m "commit de teste"
[master (root-commit) edfbe58] commit de teste
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 arquivo.txt

mingw64@windows-PC MINGW64 /c/agosti (master)
$ git status
On branch master
nothing to commit, working directory clean

mingw64@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

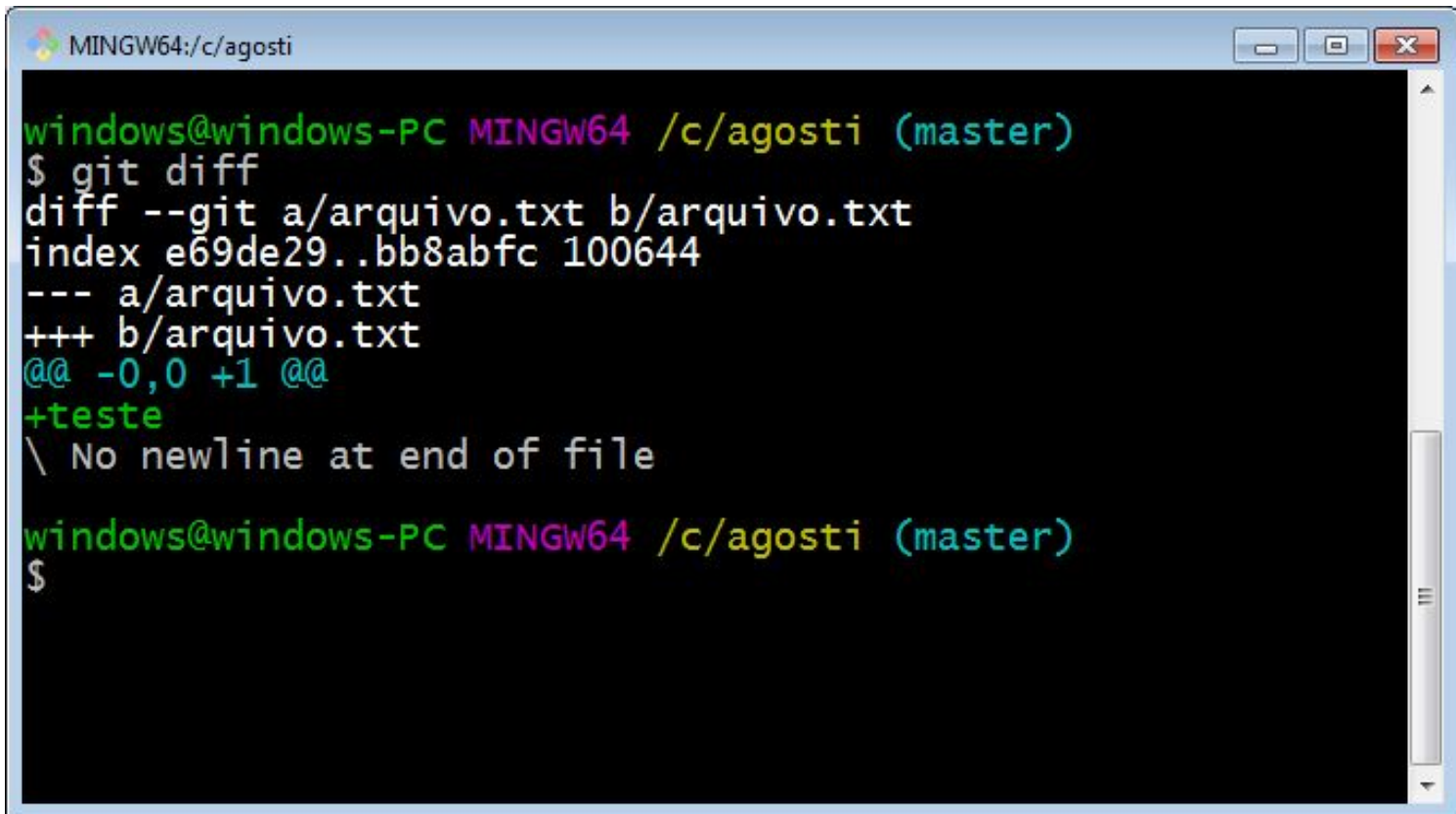
```
*.log  
build/  
temp-*
```

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)
$
```

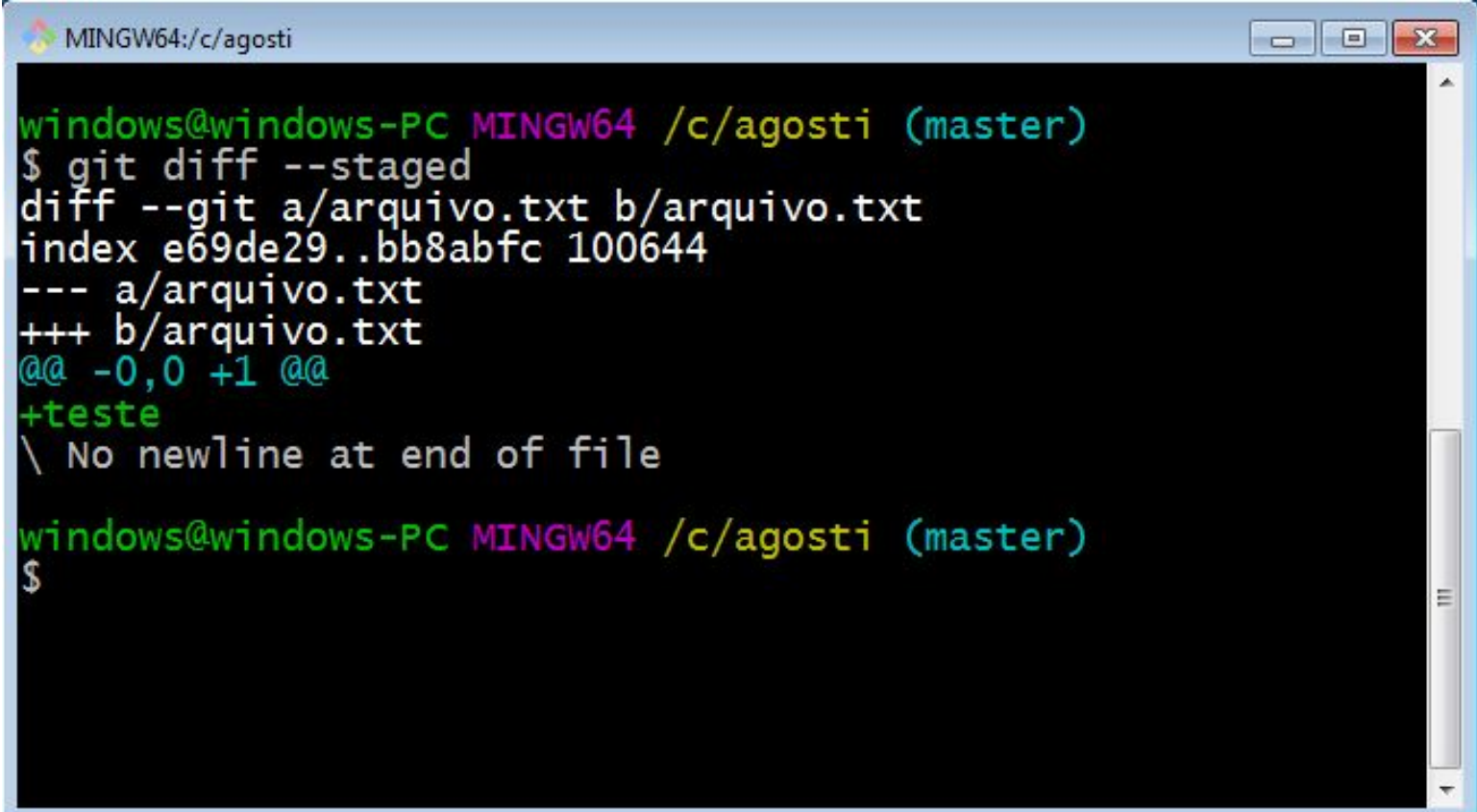

git diff

`git diff [primeiro-branch] [segundo-branch]`

mostra a diferença de conteúdo entre dois branches

git diff --staged

Mostra a diferença entre arquivos preparados e suas últimas versões



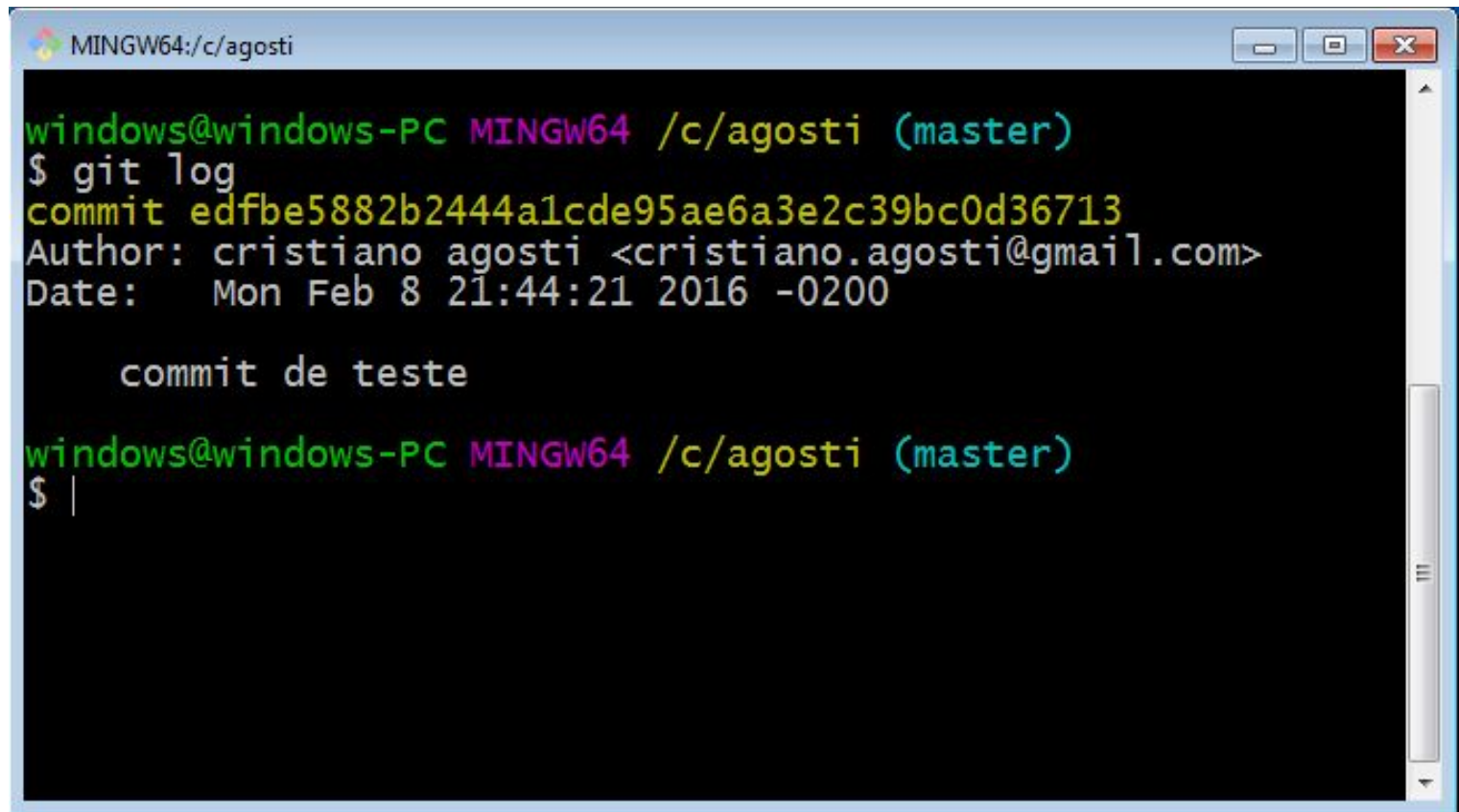
```
MINGW64:/c/agosti

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

A screenshot of a MINGW64 terminal window. The title bar shows 'MINGW64:/c/agosti'. The terminal output shows the command 'git log' being executed, resulting in a commit hash, author information, date, and a commit message. The prompt then returns to the shell.

```
windows@windows-PC MINGW64 /c/agosti (master)
$ git log
commit edfbe5882b2444a1cde95ae6a3e2c39bc0d36713
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 log --pretty=oneline

mostra em uma linha

git show [commit]

mostra mudanças de metadata e conteúdo para o commit especificado

git blame

Listagem de modificações de um arquivo

```
git blame nome-do-arquivo
```

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:

- a) Criar novo arquivo chamado `renomear_sobrenome.txt`, enviar para o repositório, renomear para `renomeado_sobrenome.txt`, enviar para o repositório
- b) Criar novo arquivo chamado `alterar1.sobrenome.txt`, enviar para o repositório, alterar o conteúdo, voltar a alteração
- c) 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
- d) Criar um arquivo compactado com extensão zip, criar arquivo `gitignore` ignorando esse arquivo, enviar para o repositório
- e) 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 resolvê-lo

A screenshot of a MINGW64 terminal window. The title bar shows 'MINGW64:/c/agosti'. The terminal output shows a Git merge command being executed, which results in a conflict in 'arquivo.txt'. The message states: 'CONFLICT (content): Merge conflict in arquivo.txt. Automatic merge failed; fix conflicts and then commit the result.' The terminal then shows the state as '(master|MERGING)' and a prompt character '\$'.

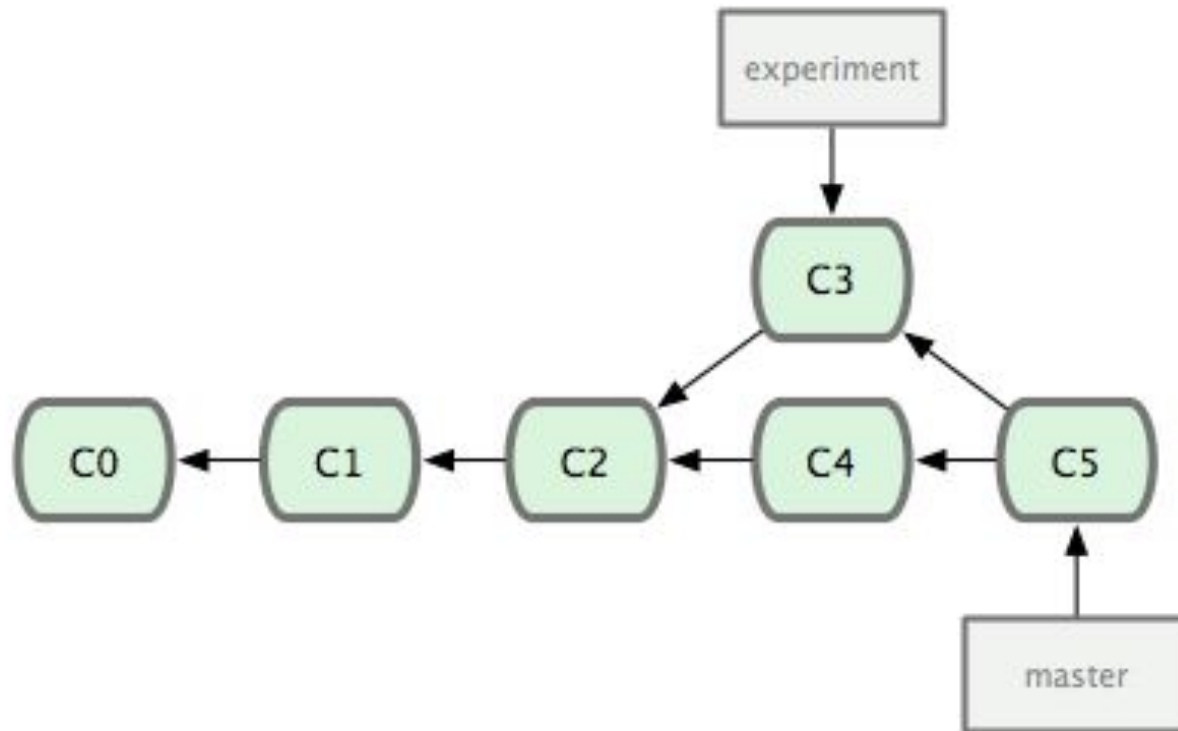
Resolução de Conflito

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

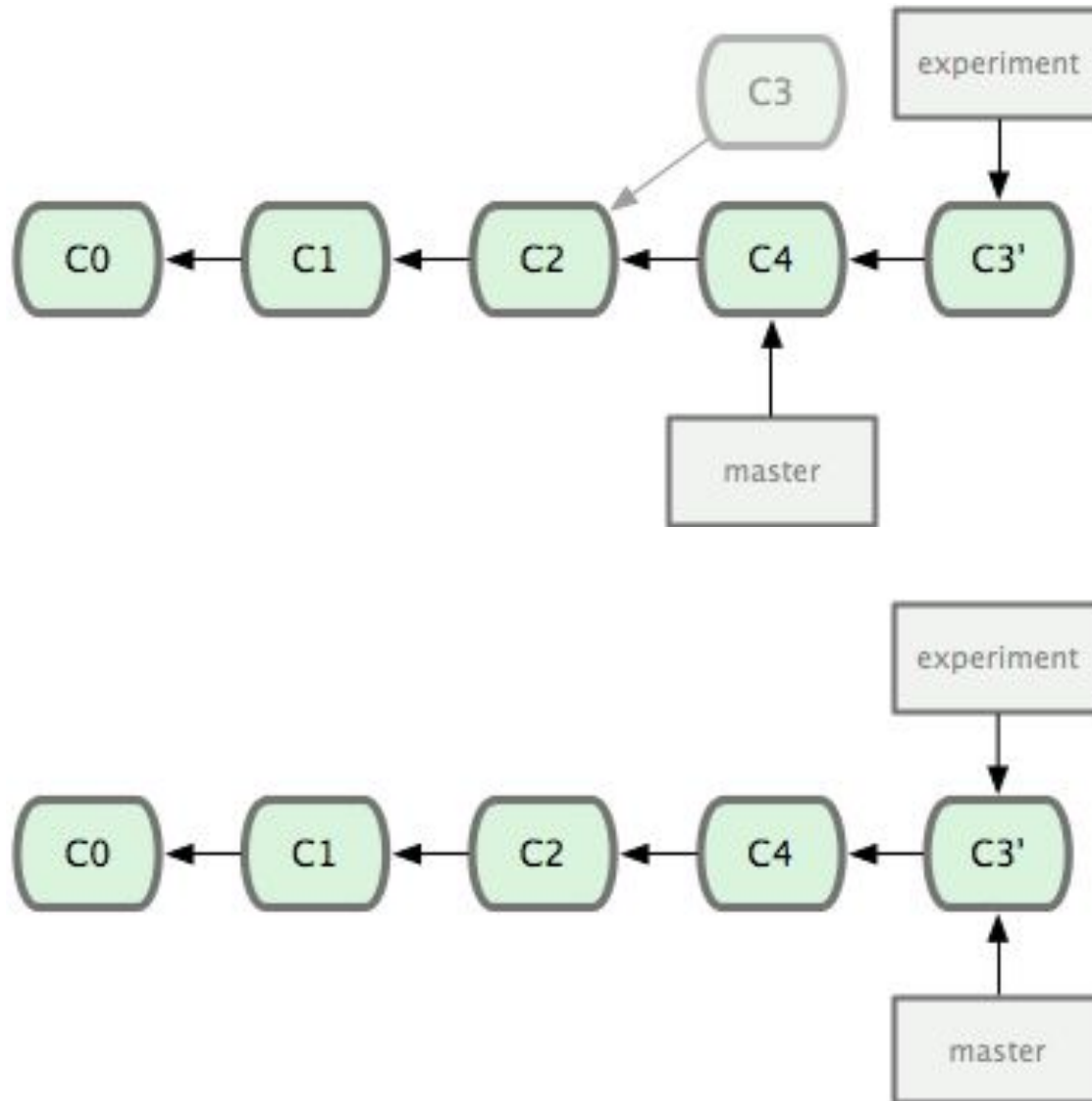
}

>>>>>>> teste
```

Merge



Rebase

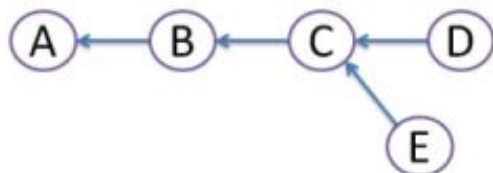


Diferença entre Merge e Rebase

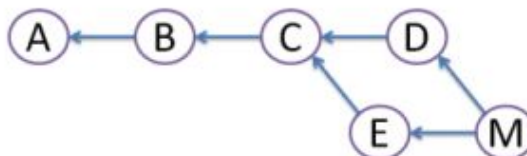
Considerando que haja a princípio 3 commits, **A**, **B** e **C**:



Em seguida o desenvolvedor José criar o commit **D** e o desenvolvedor Manuel o commit **E**:

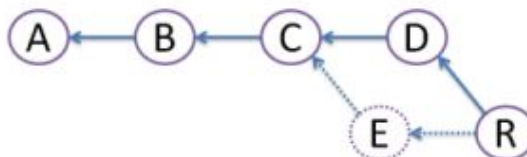


Merge



Como podemos observar, ao utilizar o Merge um novo commit **M** é criado, e ambos os commits **D** e **E** continuam no mesmo lugar. Isso acaba criando uma forma de *diamante* que confunde bastante quando vamos olhar os commits (além de criar um commit a mais).

Rebase



Ao utilizar o Rebase, um novo commit **R** com conteúdo idêntico ao do commit **M** anterior é criado, mas eliminamos o commit **E**. Com isso o histórico fica mais linear e fácil de ler e com um commit a menos.

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

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

Github

Rede Social para desenvolvedores

github.com

Criar conta

Chave Acesso



Acesso Negado

```
$ git clone git@github.com:agosti/teste.git
Cloning into 'teste'...
The authenticity of host 'github.com (192.30.253.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.113' (RSA) to the list of known hosts.
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

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's fingerprint is:
SHA256:Lyd0LTdsQ9c9X5ivC1eBpUkaRg7k5Spr4ldsKRsyysQ windows@
windows-PC
The key's randomart image is:
+---[RSA 2048]---+
|      .o.= . .   |
|      . * + = +.  |
|      . + = =. +  |
|   E . . + . o+  |
|   +..S.o.B . o  |
|   =+oo=+ + .    |
|   +oo*+ . .     |
|   o .o+ o .     |
|   ..            |
+-----[SHA256]-----+

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

id_rsa.pub

ssh-rsa


AAAAB3NzaC1yc2DAQAABAAABAQDE/OqUCyeXE4ZVZ7nO
TdYJlgeO3xzNPOEKPUI9NGzHZ9R9T5aYD2jFUewPdo+Pz
oEoQDvLK6jg7jyShmiD+OQ2717IyK6cDac3OkcFdIp8bMHL
Audxxf7sjhFt+WWIL6vmJS4Jpe9DOllJ7xPHx+KzU/WBHCclx
Ves8lBUtJkkg6/t1UMtAOomPWcRDULliRFqt71mdf3c4lUsEE
Y6tTnZZwut9LHDUbq/K2gV3OcdclJR6j79mNcVLxQjBB9o30
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

SSH keys New SSH key

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




agostl

17:e7:e1:e8:68:80:4d:5a:c8:25:d2:57:39:02:4a:2c

Added on Jun 1, 2012 — ⓘ No recent activity

Delete



agostimac

eb:4a:1e:93:0d: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

 agosti / aulas-git

 Unwatch ▾ 1

 Star 0

 Fork 0

 Code

 Issues 0

 Pull requests 0

 Wiki

 Pulse

 Graphs

 Settings

Repositório de Códigos do curso de Git da Unoesc — Edit

 1 commit

 1 branch

 0 releases

1 contributor

Branch: master ▾

New pull request

New file

Find file

HTTPS ▾

https://github.com/agos



Download ZIP



agosti Initial commit

Latest commit fcec360 just now

 README.md

Initial commit

just now

 README.md

aulas-git

Repositório de Códigos do curso de Git da Unoesc

Clonando Repositório do Github

(usando ssh)

git clone [git@github.com:agosti/aulas-git.git](ssh://git@github.com:agosti/aulas-git.git)

(usando https)

git clone https://github.com/agosti/aulas-git.git

Informando um Repositório Remoto

Adicionando

git remote add origin git@github.com:agosti/projeto.git

Setando

git remote set-url origin git@github.com:agosti/projeto.git

Pull e Push

git pull [alias remoto] [branch]

busca mudanças

git fetch [alias remoto] [branch]

busca mudanças sem commit

git push [alias remoto] [branch]

envia as mudanças para o servidor remoto

git pull origin master --allow-unrelated-histories (relacionado
branches remoto e local)

Prática 4

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

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
git remote add origin git@github.com:agosti/projeto.git

https://try.github.io

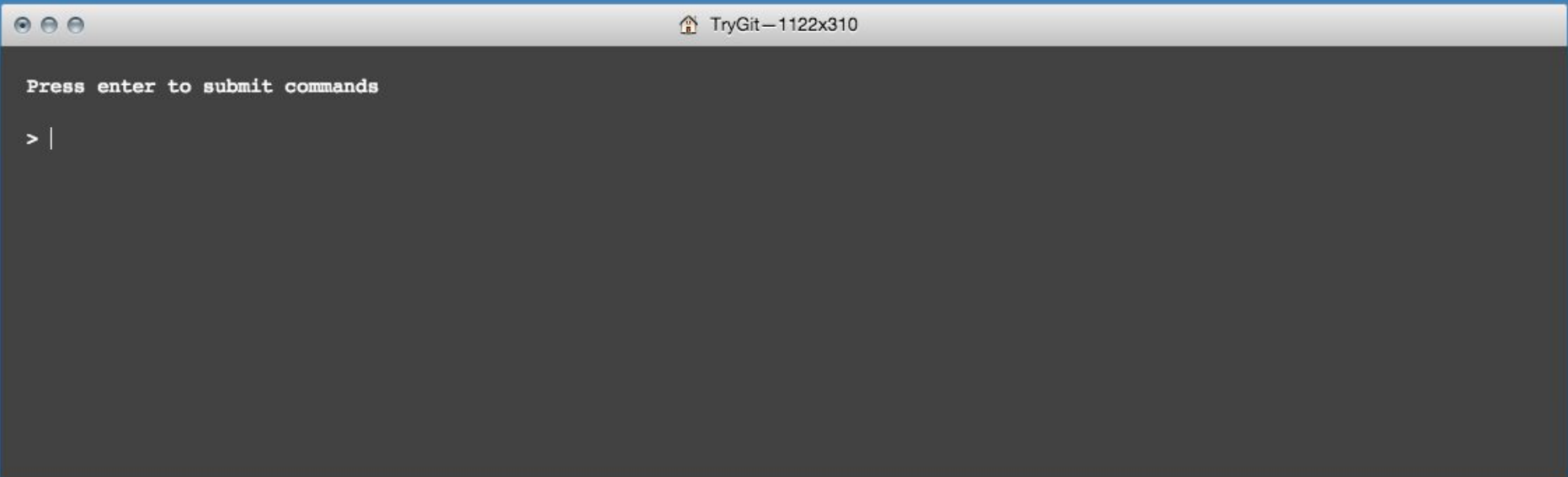


1.1 · Got 15 minutes and want to learn Git?

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`



Prática 5 - <https://try.github.io>

- a) Refazer os passos do try.github.io em um repositório remoto real

Git 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

Guardar para fazer depois

Git 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`

Prática 6

- a) Acessar a branch b_stash
- b) Fazer várias modificações
- c) Criar stash
- d) Verificar
- e) Voltar stash
- f) Verificar

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

Prática 7

a) Criar Alias para os comandos

status

checkout

commit

Git cherry-pick

Trazer apenas alguns commits para o branch atual

```
git cherry-pick 19f0bb7d8b4be8ecd687b48fca301b71b95ea
```

Cuidar ao usar pois podem haver commits dependentes!!!!

Prática 8

- a) Criar um novo branch b_cherry_pick
- b) Acessar o branch novo
- c) Realizar 3 commits de de mudanças quaisquer
- d) Retornar ao master
- e) Buscar o commit 2

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

[Aprenda Git](#)[Login](#)[Sobre](#)[Ajude-nos no GitHub](#)

Aviso! Estamos em Beta. Caso encontre algum problema, envie um email para 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](#)

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, nederlands, polski, русский, türkçe,

မြန်မာ, 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!



https://services.github.com/on-demand/downloads/pt_BR/github-git-cheat-sheet.pdf

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

Prática 9

- a) Clonar projeto do github do Professor
- b) Criar novo arquivo sobrenome.txt, enviar para o servidor
- c) Simular conflitos entre colegas

Url do Projeto

[git@github.com](https://github.com/agosti/aulagit.git):agosti/aulagit.git

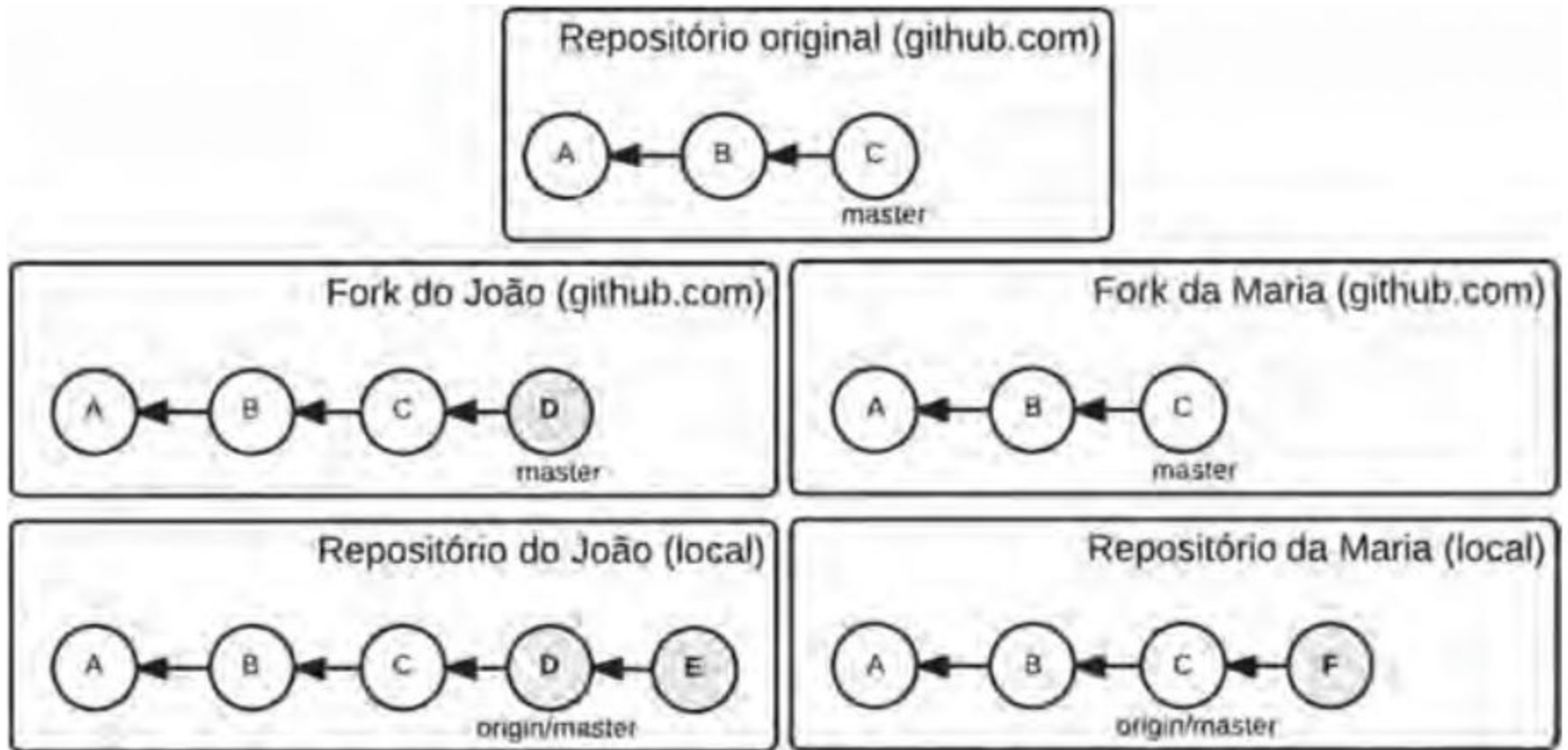
<https://github.com/agosti/aulagit.git>

Colaborar com Projetos Open Source


Fork clonar para outro repositório

Pull Request enviar as modificações

Fork e Pull Request



Fork e Pull Request

 This repository Search Pull requests Issues Marketplace Explore

agosti / octobox

Unwatch 1 Star 0 Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings


teste git

Add topics


Edit

9 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download


 Cristiano Agosti Merge branch 'master' of <https://github.com/agosti/octobox>

Latest commit 79e5c16 2 hours ago

 README.md

Initial commit

3 hours ago

 README.md

octobox

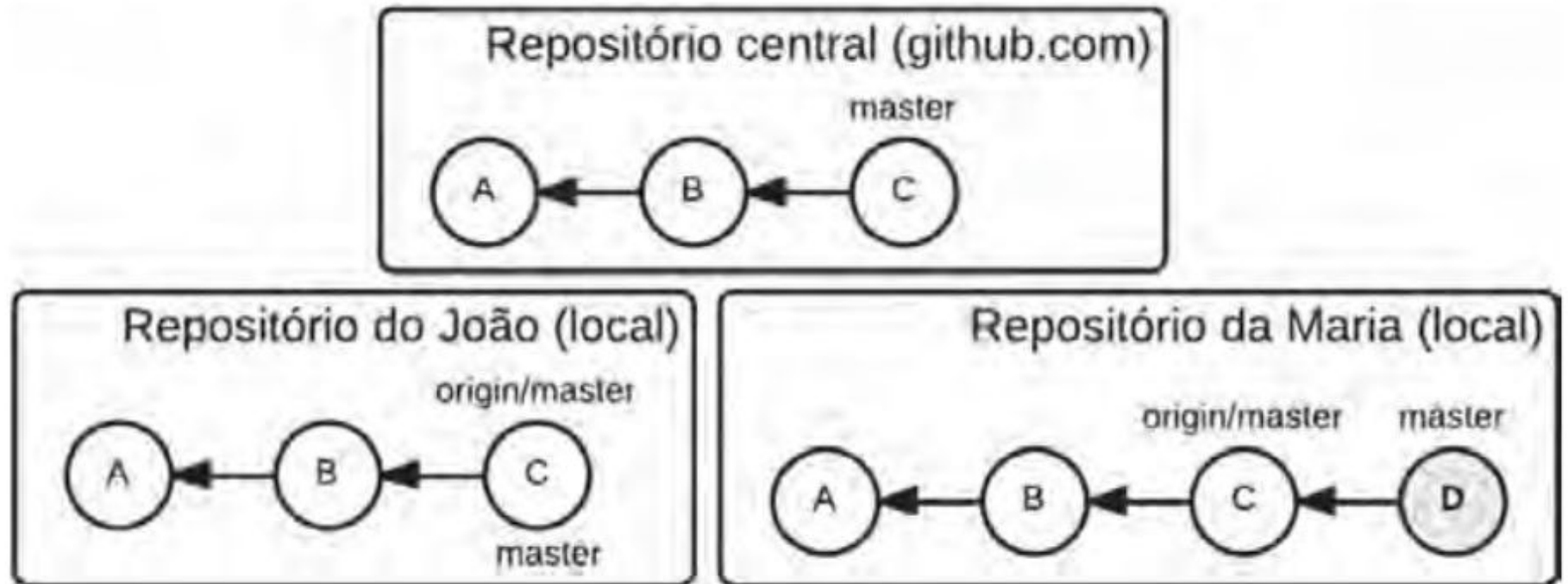
teste git

Prática 10

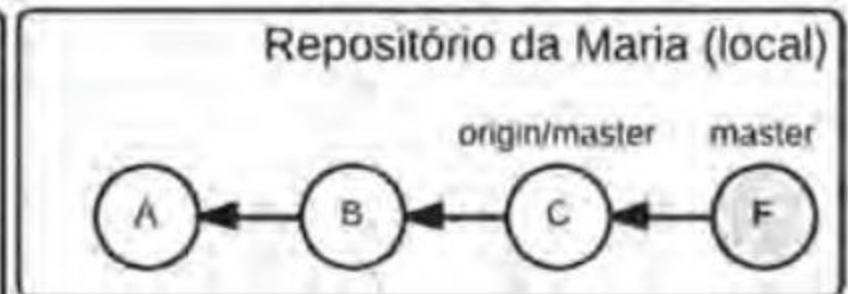
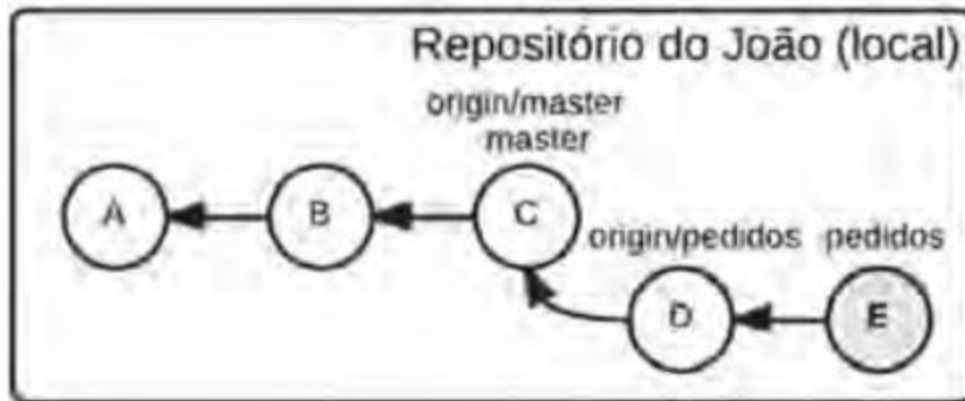
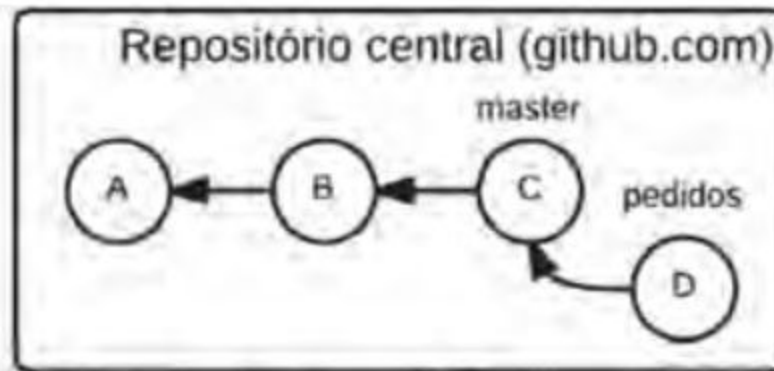
- a) Criar o Fork de um projeto
- b) Modificar
- c) Enviar Fork
- d) Realizar o Pull Request

Estratégias

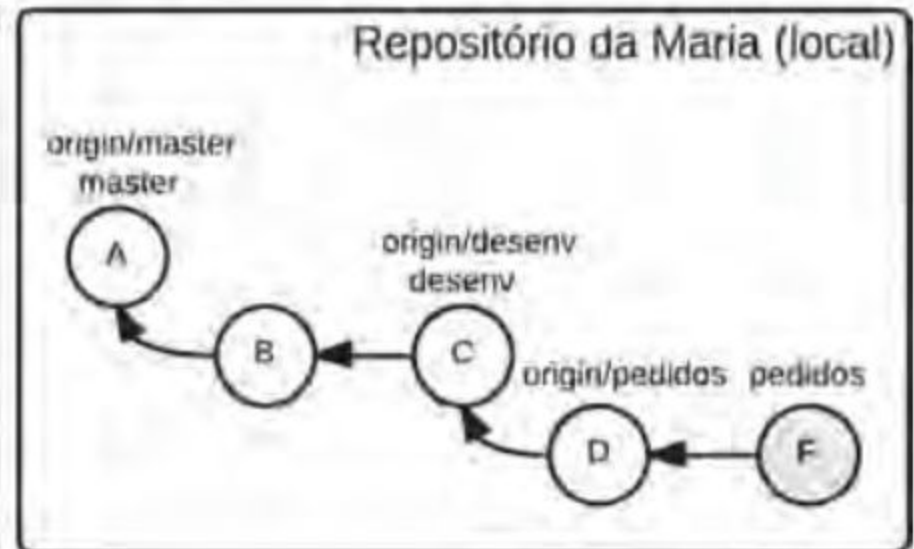
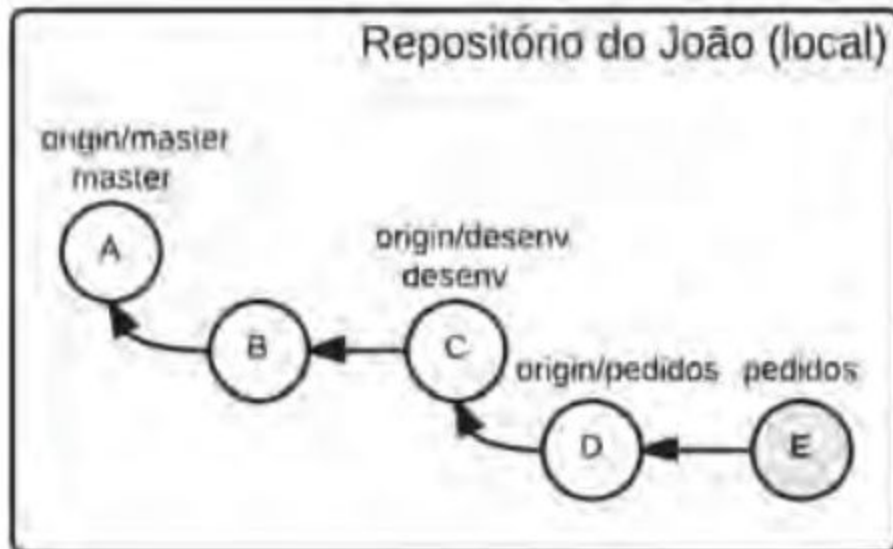
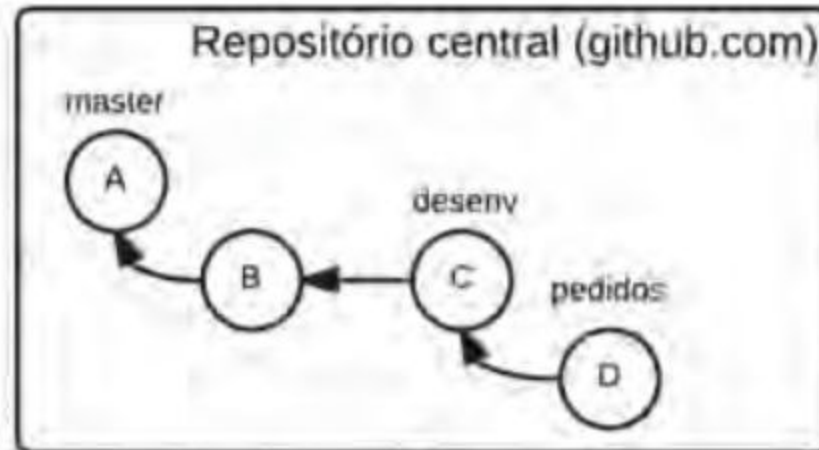
Utilizando só a branch master com um repositório central



Utilizando branches por funcionalidade com um repositório central

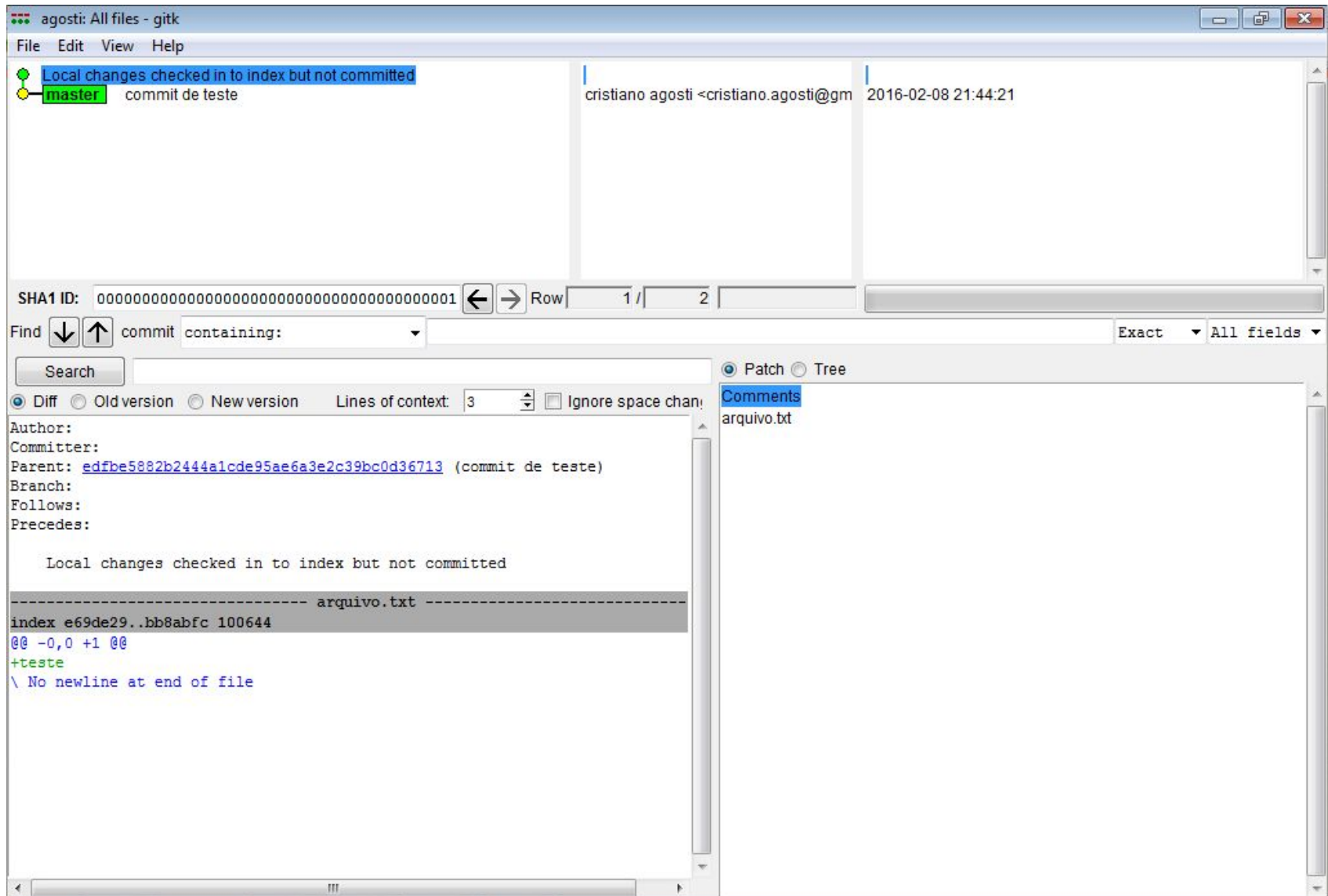


Utilizando branches por etapa de desenvolvimento com um repositório central

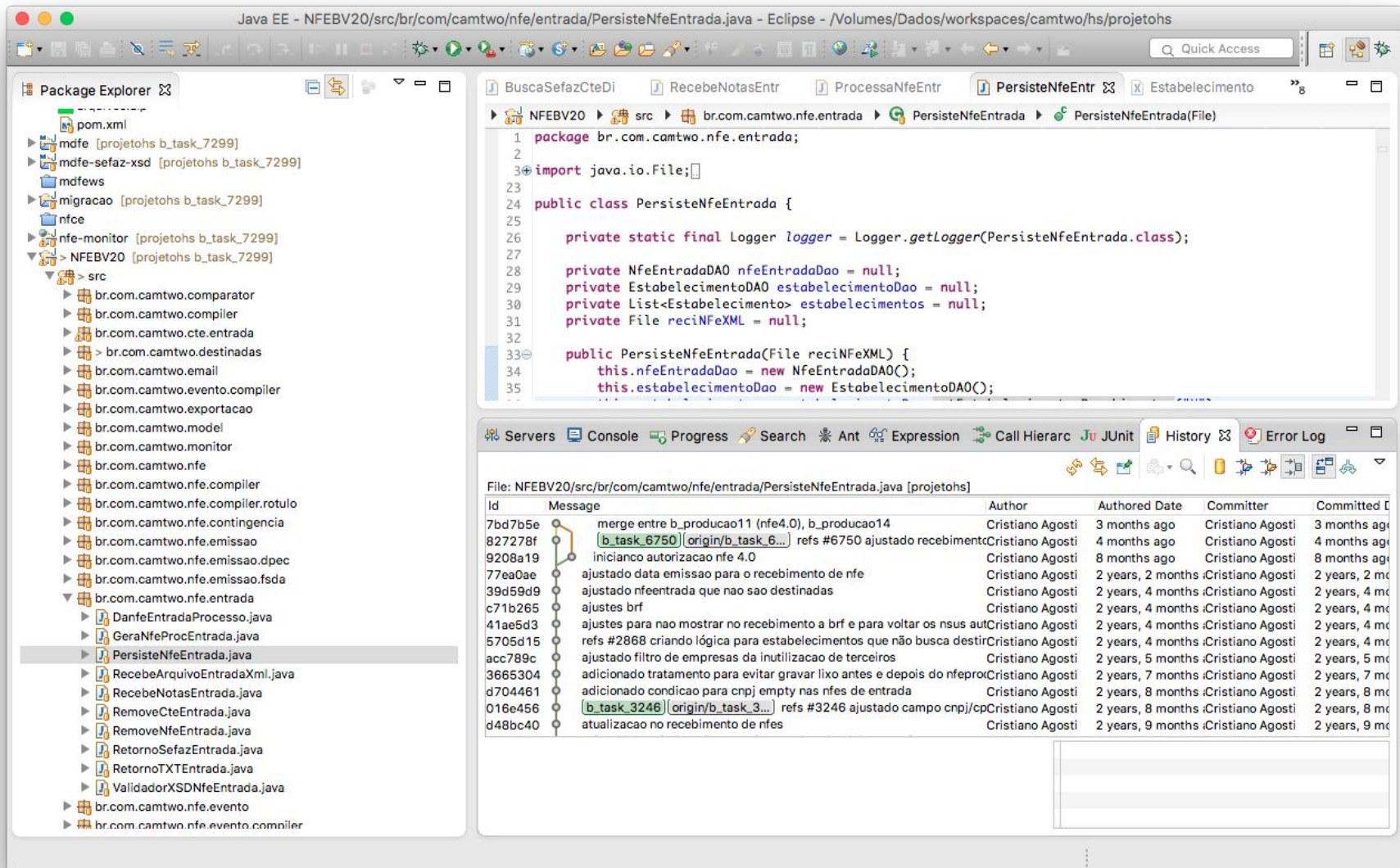


Ferramentas Gráficas

gitk



Eclipse



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



A free Git & Mercurial client for Windows or Mac.

Download SourceTree Free

🍏 Mac OS X 10.10+

🖥️ Download SourceTree for Windows 7+

A screenshot of the SourceTree application interface on a Mac. The window title is "sourcetree-assets (Git)". The top toolbar includes icons for Commit, Pull, Push, Fetch, Branch, Merge, Stash, Show in Finder, Terminal, and Settings. The left sidebar shows a "WORKSPACE" section with "File status", "History", and "Search", and a "BRANCHES" section with "master" selected. The main area displays a commit history table with columns for Graph, Commit, Author, Description, and Date. The commit "86463b7" is highlighted in blue.

Graph	Commit	Author	Description	Date
	0de7467	Mike Minns...	Merge branch 'master' of https://stash.atlassian.com/scm/srctree/sourcetree...	Feb 4, 2016, 1:53...
	1668ff8	Mike Minns...	Added standalone copy of the clone_new for the Windows toolbar	Feb 4, 2016, 1:52...
	406c23d	Joel Unger <...>	Adding sidebar folder tiff	Feb 4, 2016, 3:52...
	7a7f37c	Joel Unger <...>	Fix bb sizes ... again	Feb 3, 2016, 2:04...
	f99d729	Joel Unger <...>	Updated bitbucket and github icon sizes and selected state.	Feb 3, 2016, 1:36...
	86463b7	Mike Minns...	Release 1.0.21 Metadata	Feb 2, 2016, 3:31...
	d799d45	Mike Minns...	Resized wizard navigation panel. Picked up latest icons	Feb 2, 2016, 3:28...

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



Products

Downloads

Support

Store

Company



DIFFMERGE

Overview

Documentation

Register

Release Notes

System Requirements

Support Forum

License Agreement

Download

Product Features

DiffMerge is an application to visually compare and merge files on Windows, OS X and Linux.



Diff

Graphically shows the changes between two files. Includes intra-line highlighting and full support for editing.



Merge

Graphically shows the changes between 3 files. Allows automatic merging (when safe to do so) and full control over editing the resulting file.



Folder Diff

Performs a side-by-side comparison of 2 folders, showing which files are only present in one file or the other, as well as file pairs which are identical or different.

Windows Explorer Integration.

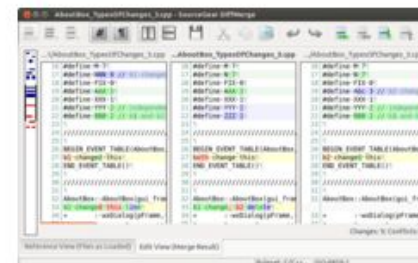
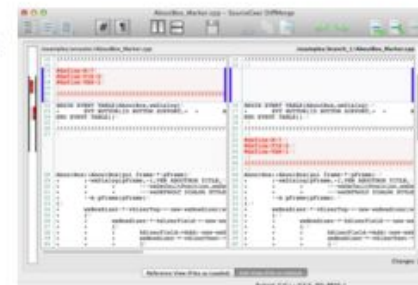
Right-click on any two files in Windows Explorer to diff them immediately.

Configurable.

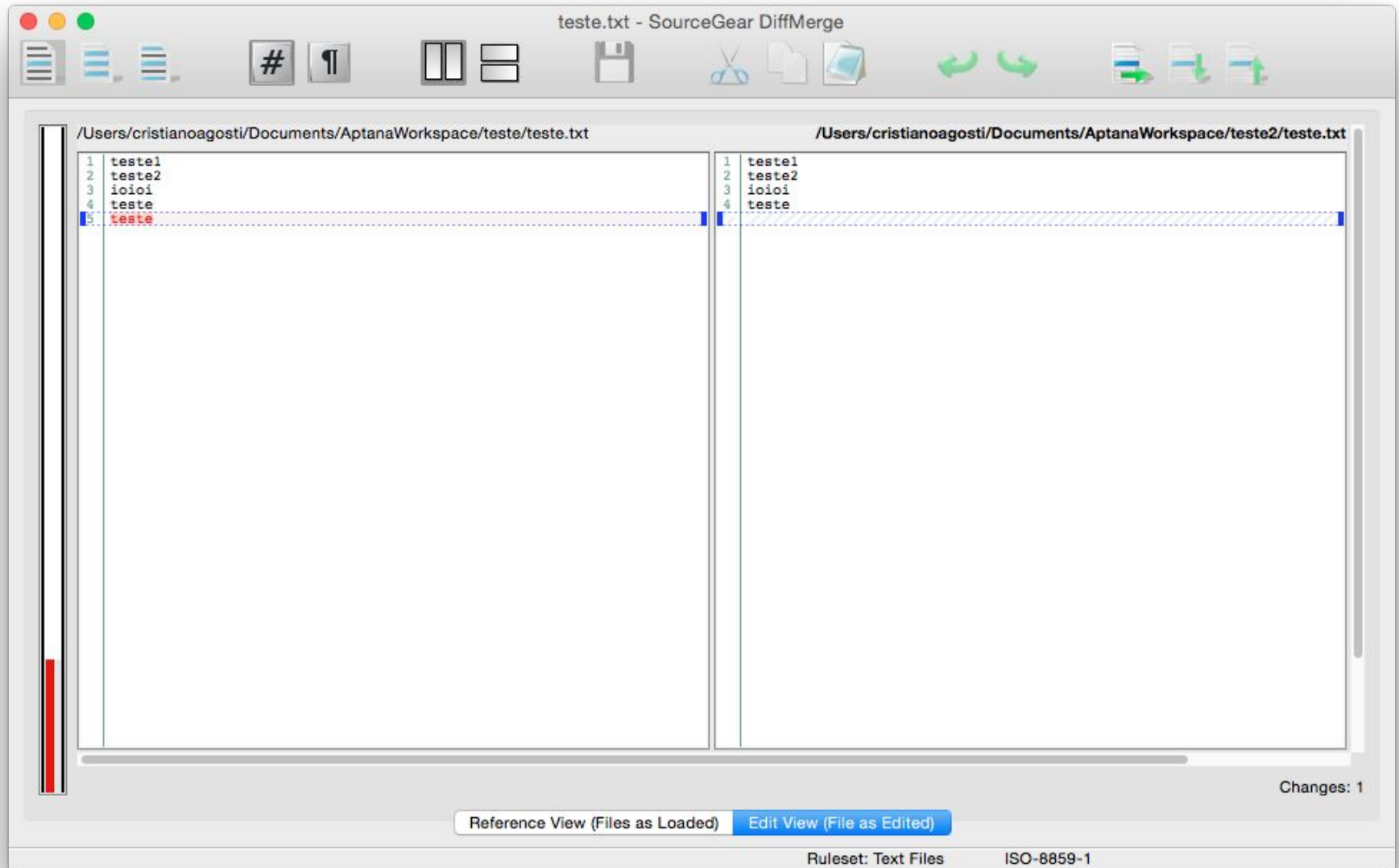
Rulesets and options provide for customized appearance and behavior.

Cross-platform.

Identical feature set on Windows, OS X, and Linux.



Diff Merge



Source Tree

The screenshot displays the Source Tree application window for a repository named 'teste (Git)'. The interface includes a top toolbar with various Git actions like Commit, Checkout, Reset, Stash, Add, Remove, Add/Remove, Fetch, Pull, Push, Branch, Merge, and Tag. A left sidebar shows the 'FILE STATUS' with 'Working Copy' checked, and sections for 'BRANCHES' (showing 'master'), 'TAGS', 'REMOTES' (showing 'origin'), 'STASHES', 'SUBMODULES', and 'SUBTREES'. The main area is divided into a 'Graph' view showing a commit history and a 'Commit' details view for the selected commit 'c2ec4f4'. The commit details show the author 'Cristiano Agosti' and the date '19 de fevereiro de 2016 13:50:22 BRST'. The 'File contents' view on the right shows the file 'teste.txt' with the content 'ioioi'.

teste (Git)

View Commit Checkout Reset Stash Add Remove Add/Remove Fetch Pull Push Branch Merge Tag Show in Finder Git Flow Terminal Settings

FILE STATUS
Working Copy

BRANCHES
master

TAGS

REMOTES
origin

STASHES

SUBMODULES

SUBTREES

All Branches Show Remote Branches Ancestor Order Jump to:

Graph	Description	Commit	Author	Date
master	atualizado	e91b32f	Cristiano Agosti...	19/02/2016 14:50
origin/master	atualizacao	eb6667f	Cristiano Agosti <...	19/02/2016 14:47
origin/HEAD	teste	366f2c5	Cristiano Agosti <...	19/02/2016 14:49
	mudancas usuario 2	c6ec60e	Cristiano Agosti <...	19/02/2016 14:45
teste	teste	c2ec4f4	Cristiano Agosti <...	19/02/2016 13:50

Sorted by path

+ teste.txt

+ teste.txt

File contents Reverse hunk

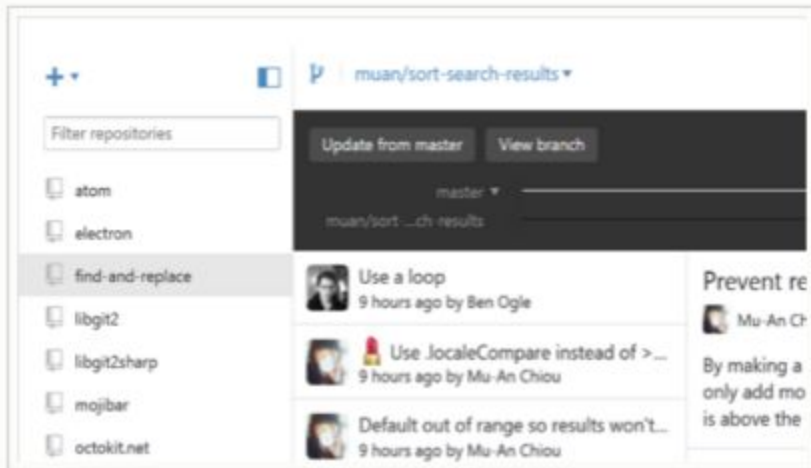
```
1 +
2 + ioioi
```

teste

Commit: c2ec4f49ab672e0e6bccf17737e937e4c25bd1be...
Parents: Cristiano Agosti <cristiano.agosti@camtwo.com....
Author: 19 de fevereiro de 2016 13:50:22 BRST
Date:

master Clean

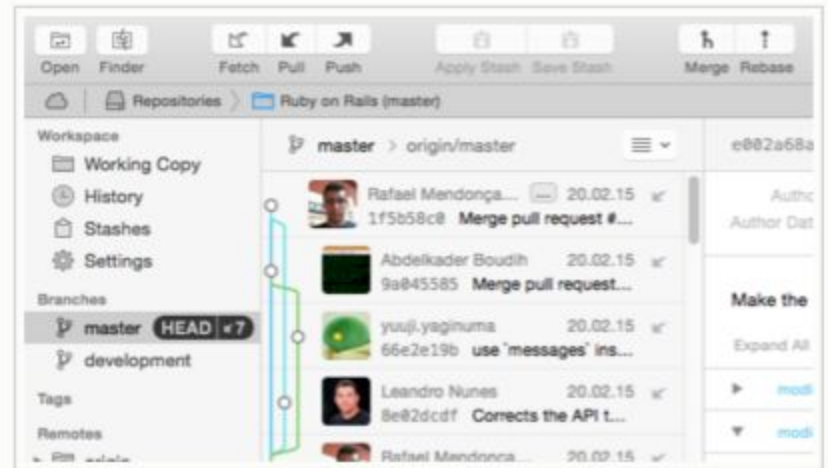
Atlassian



GitHub Desktop

Platforms: Windows, Mac

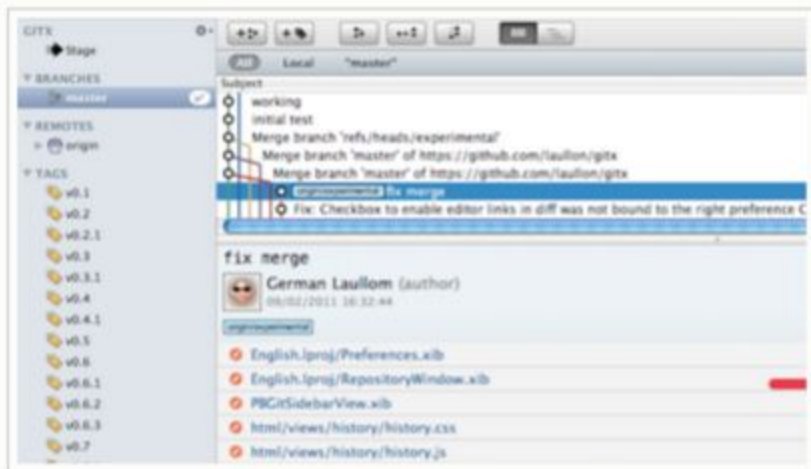
Price: Free



Tower

Platforms: Mac

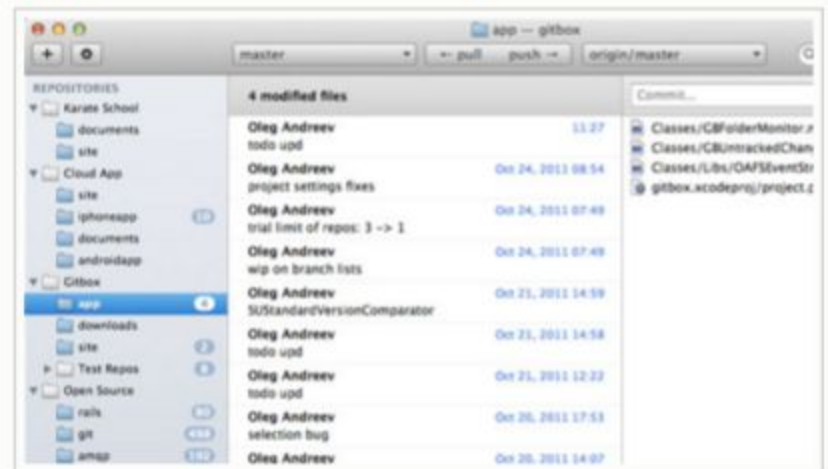
Price: \$69/user (Free 30 day trial)



GitX-dev

Platforms: Mac

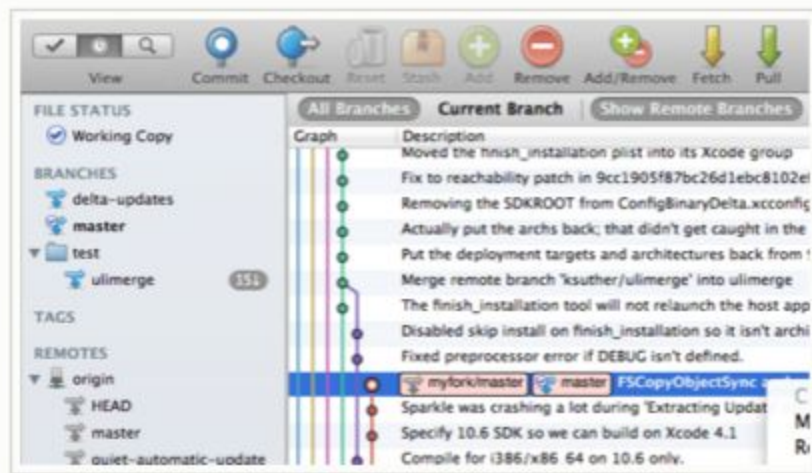
Price: Free



Gitbox

Platforms: Mac

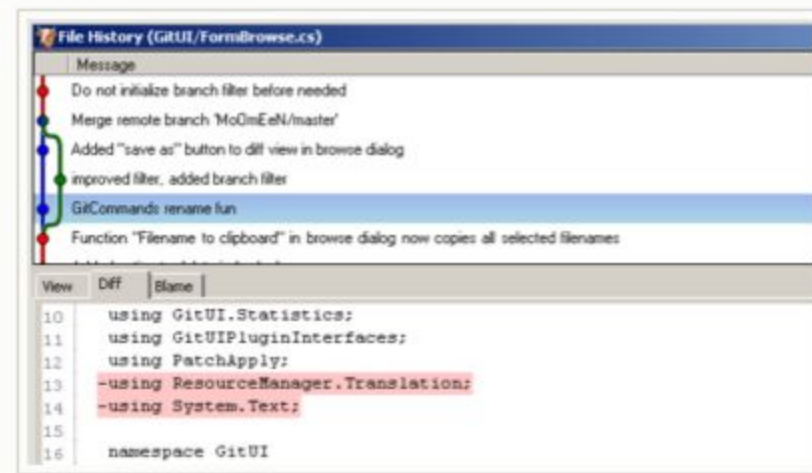
Price: \$14.99



SourceTree

Platforms: Mac, Windows

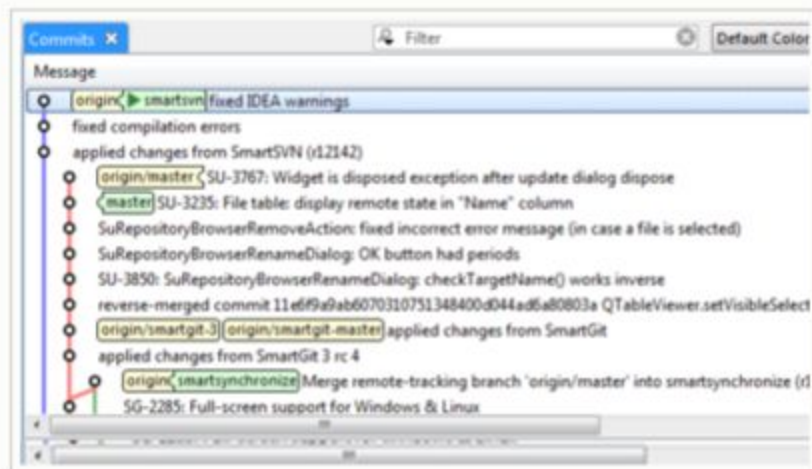
Price: Free



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

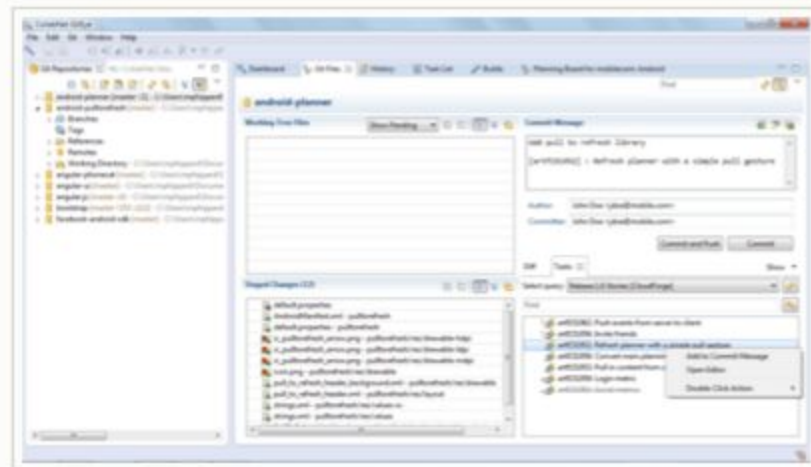
Price: Free



GitUp

Platforms: Mac

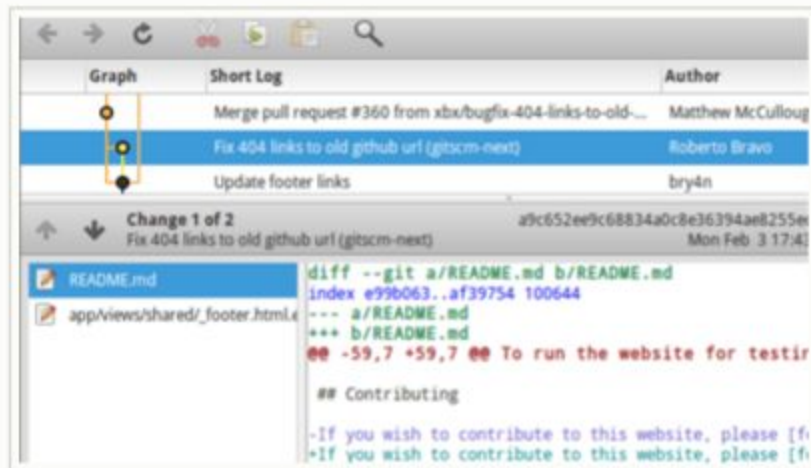
Price: Free



GitEye

Platforms: Windows, Mac, Linux

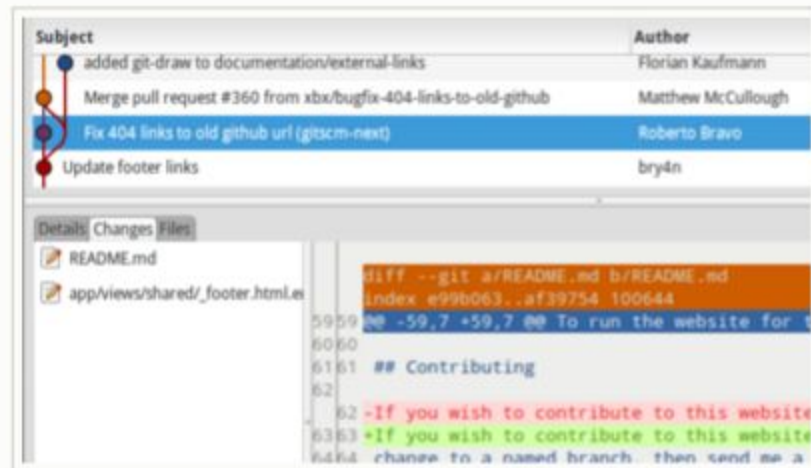
Price: Free



gitg

Platforms: Linux

Price: Free



gitg

Platforms: Linux

Price: Free

Protocolos do Git

Local (file)

Git

SSH

Http(S)

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)/'
```

```
}
```

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
```


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
- <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
- <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