

Introdução à análise geoespacial com R

1 Controle de versão, Git e GitHub

Maurício H. Vancine

Milton C. Ribeiro

19/10/2020



1 Controle de versão, Git e GitHub

Conteúdo

1. Conferir os notebooks e PCs
2. Controle de versão
3. Git
4. GitHub
5. Funcionamento do controle de versão
6. Fork
7. Iniciando: init ou clone
8. Configurando: config
9. Básico: add, status, commit e log
10. Ramificações: branch, switch e merge
11. Remoto: push e pull
12. Pull request
13. Interface Gráfica RStudio



Vamos conferir os notebooks?!

Software e Hardware



Software é aquilo que você xinga.
Hardware é aquilo que você chuta.



Café com Código

1.1 Verificar os notebooks

R (>4.0.x)

<https://www.r-project.org/>



1.1 Verificar os notebooks

RStudio (>1.3.x)

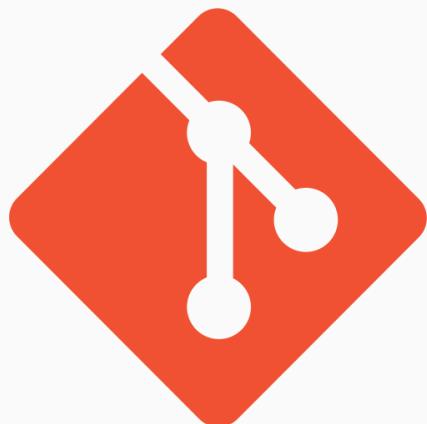
<https://www.rstudio.com/>



1.1 Verificar os notebooks

git (>2.28)

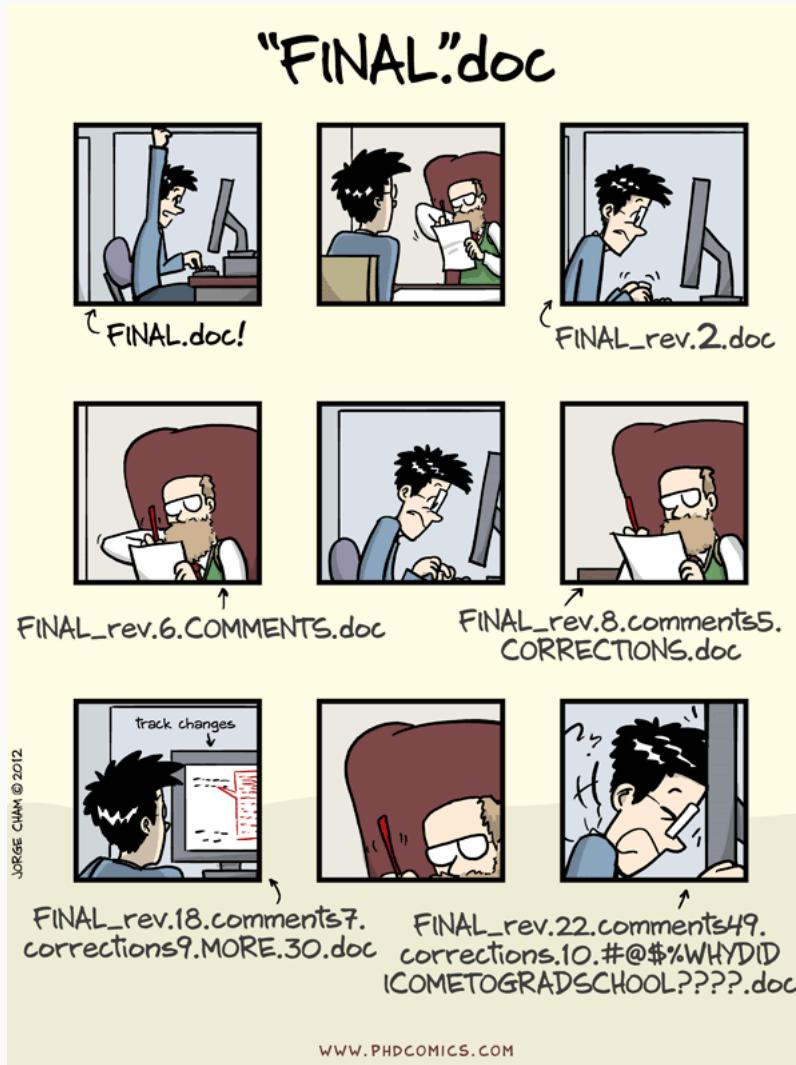
<https://git-scm.com/>



git

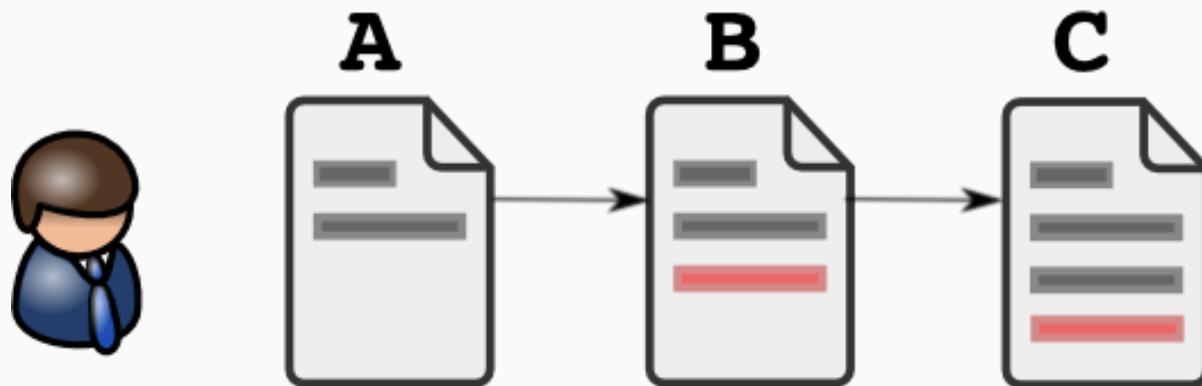
Tudo instalado? Então bora!

1.2 Controle de versão



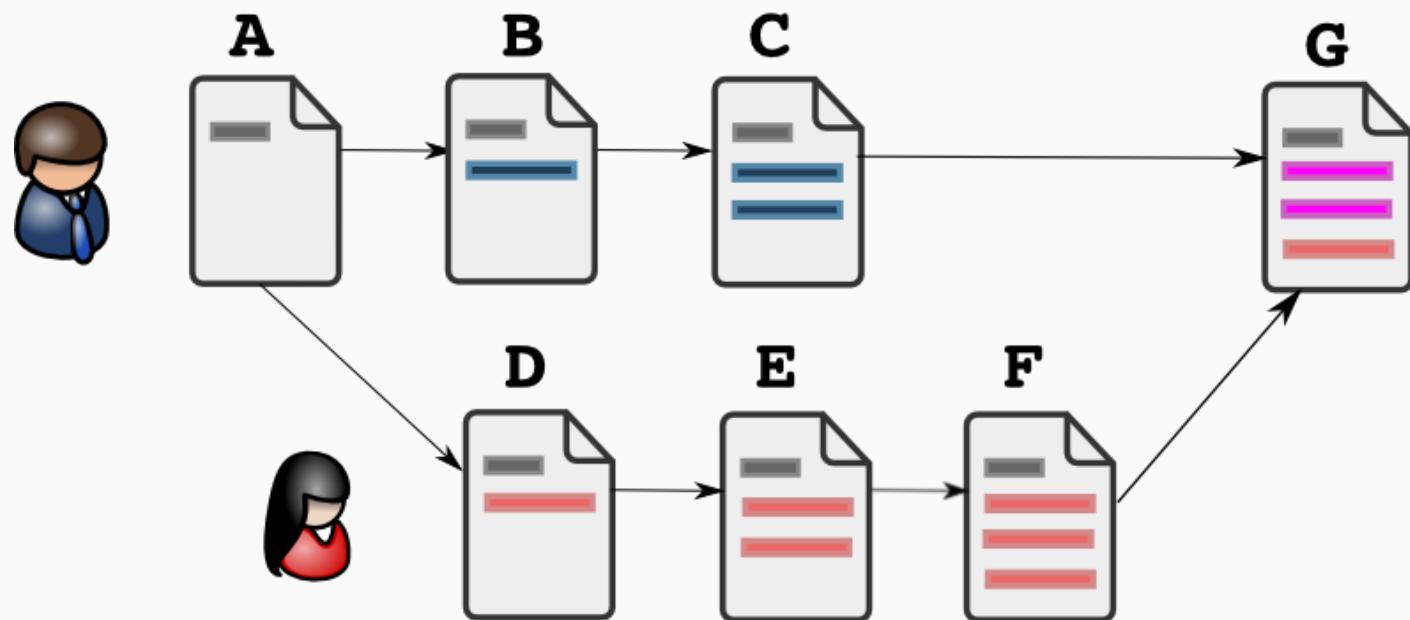
1.2 Controle de versão

Manejar projetos **individualmente**



1.2 Controle de versão

Manejar projetos **compartilhados**



1.3 Git

Software que faz o **controle de versão**

Maneja os repositórios locais (notebook) e remotos (GitHub)



1.4 GitHub

Repositório remoto

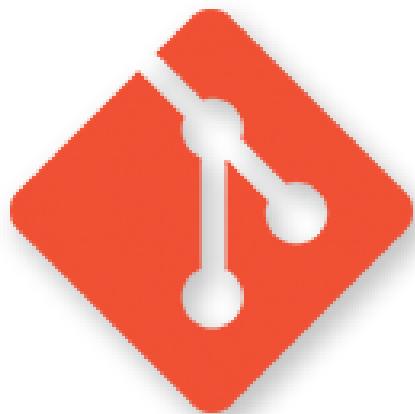
Plataforma de hospedagem de código-fonte com controle de versão usando o Git



ATENÇÃO!

Git: software que faz o controle de versão

GitHub: repositório remoto que hospeda os arquivos versionados



git

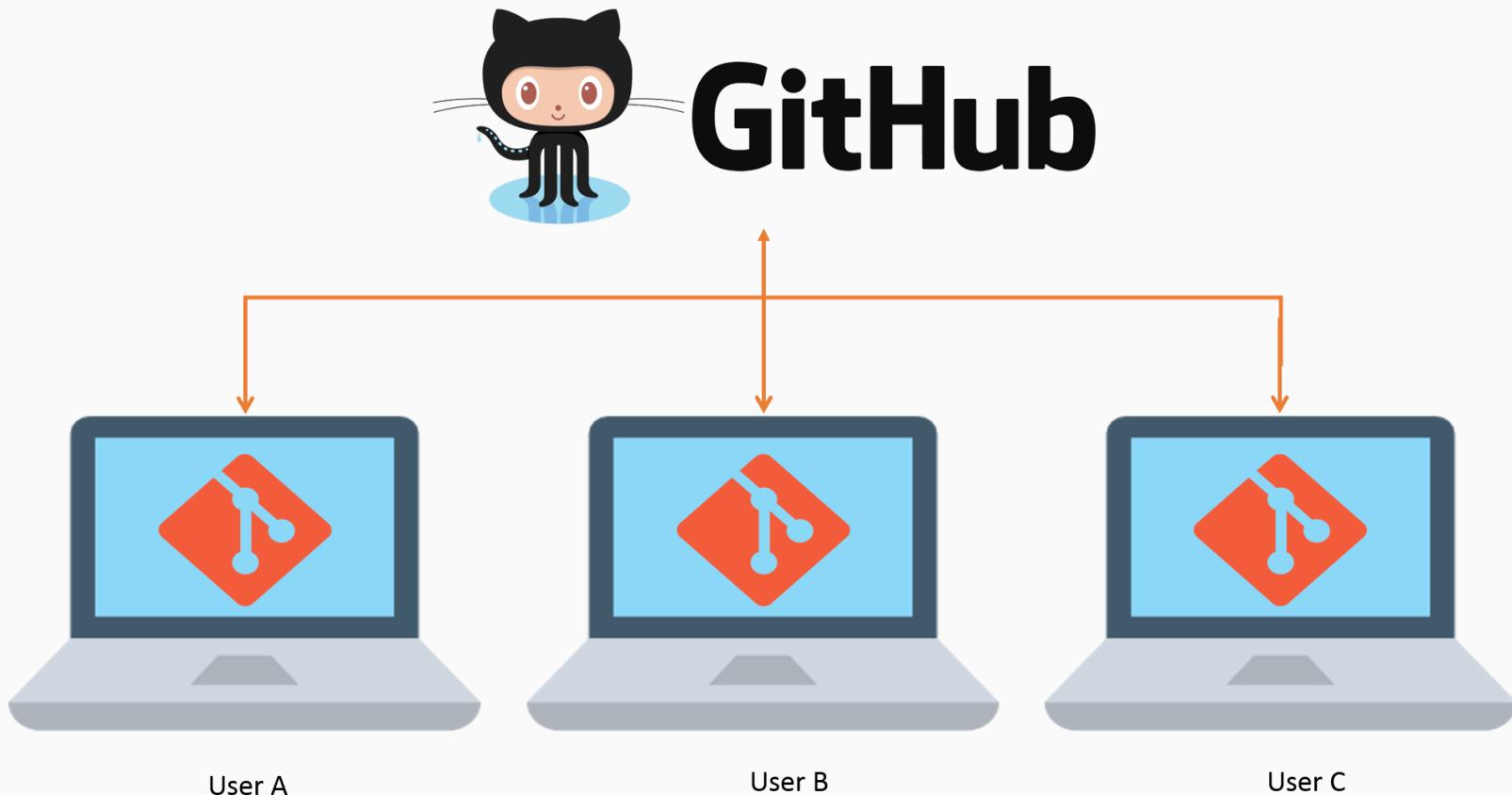


github
SOCIAL CODING

ATENÇÃO!

Git: software que faz o controle de versão

GitHub: repositório remoto que hospeda os arquivos versionados



1.4 GitHub

Vamos **criar uma conta no GitHub** (caso não possuam)

<https://github.com>

1.4 GitHub

Salvem ou recuperem essas informações!!!

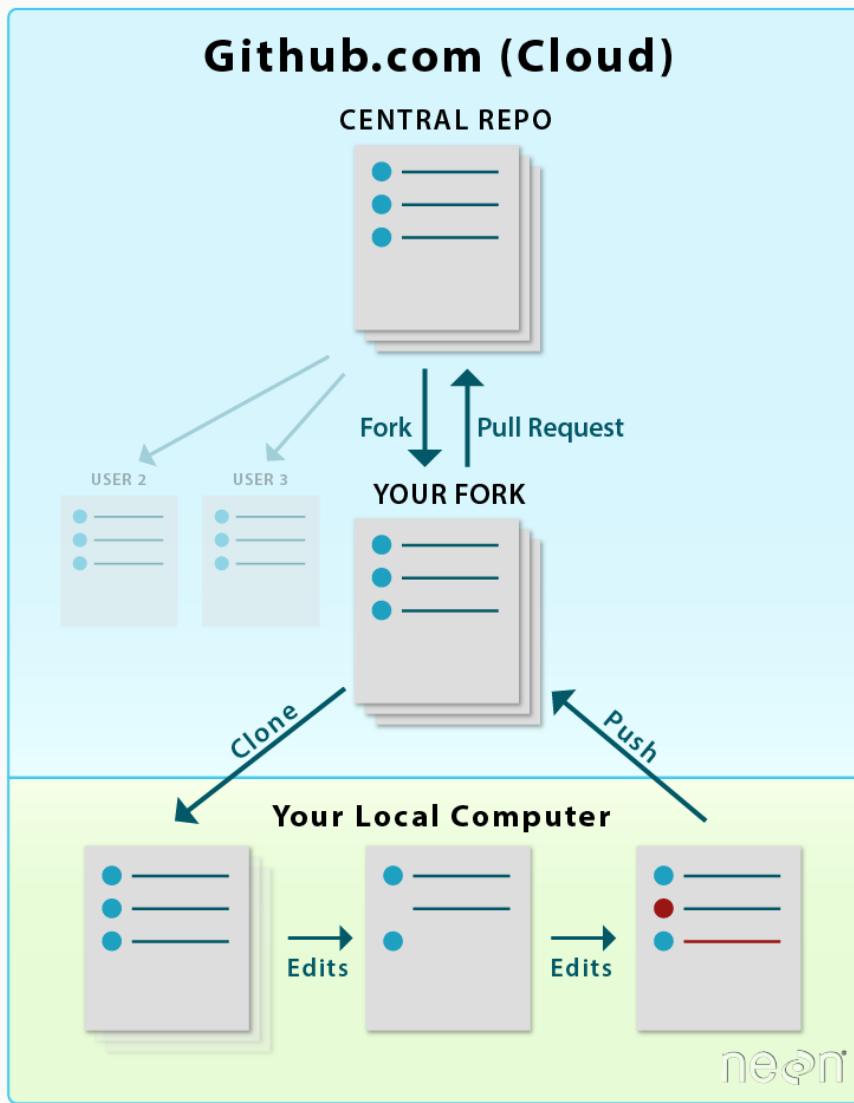
username: mauriciovancine

email: mauricio.vancine@gmail.com

senha: !@#\$%&*+

Tudo bem até aqui?

1.5 Funcionamento do controle de versão



THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

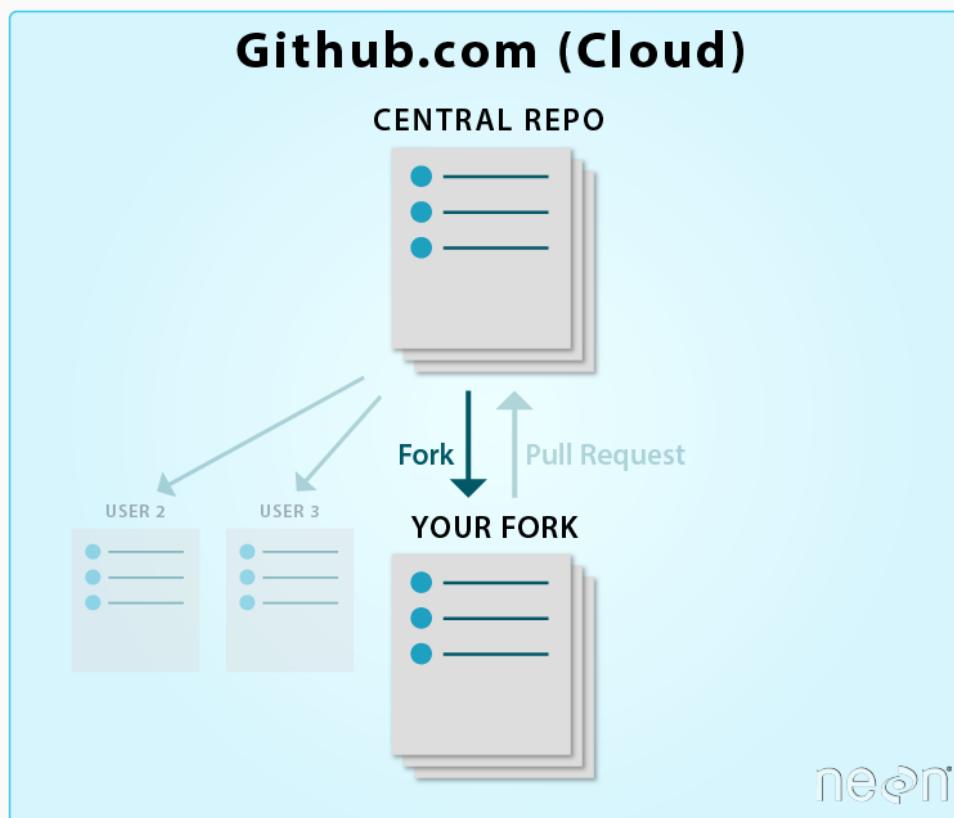
COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



1.6 Fork

Fork: copia um repositório remoto para o seu GitHub



1.6 Fork

Forken o repositório da disciplina para o GitHub de vocês

<https://github.com/mauriciovancine/disciplina-analise-geoespacial-r>

The screenshot shows a GitHub repository page for 'mauriciovancine / disciplina-analise-geoespacial-r'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. On the right side, there are buttons for Unwatch (1), Star (0), and Fork (1). The 'Fork' button is highlighted with a red box. Below the navigation, the repository details show 'master' branch, '1 branch', '0 tags', and a commit from 'mauriciovancine' on '2020-10-07' with commit hash '83641ff'. The file list includes '00_plano_ensino', '01_aulas', '02_scripts', '03_dados', '04_exercicios', '.gitignore', 'README.md', and 'disciplina-geoprocessamento.Rproj'. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published' and 'Create a new release'. The 'Packages' section says 'No packages published' and 'Publish your first package'.

mauriciovancine / **disciplina-analise-geoespacial-r**

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

mauriciovancine 2020-10-07 83641ff 11 minutes ago 3 commits

File	Created	Last Commit
00_plano_ensino	2020-10-07	11 minutes ago
01_aulas	2020-10-07	11 minutes ago
02_scripts	2020-10-07	11 minutes ago
03_dados	2020-10-07	11 minutes ago
04_exercicios	2020-10-07	11 minutes ago
.gitignore	2020-10-07	11 minutes ago
README.md	Initial commit	6 months ago
disciplina-geoprocessamento.Rproj	2020-10-07	11 minutes ago

About

No description, website, or topics provided.

Readme

Releases

No releases published

Create a new release

Packages

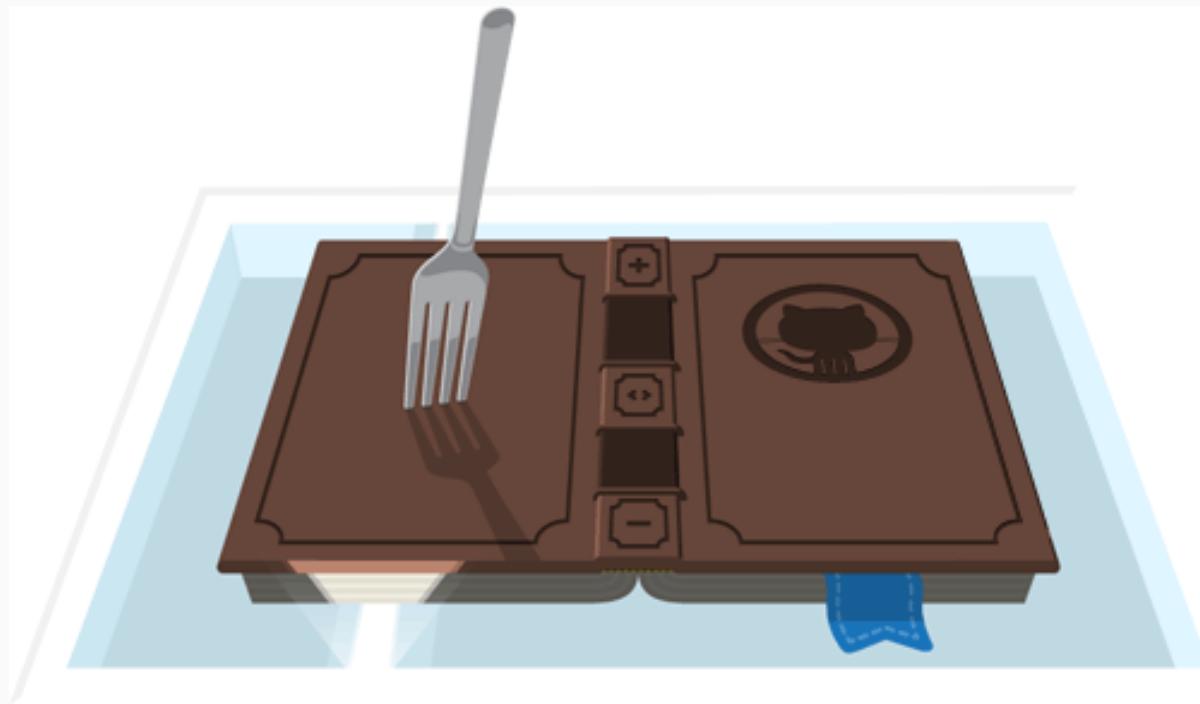
No packages published

Publish your first package

1.6 Fork

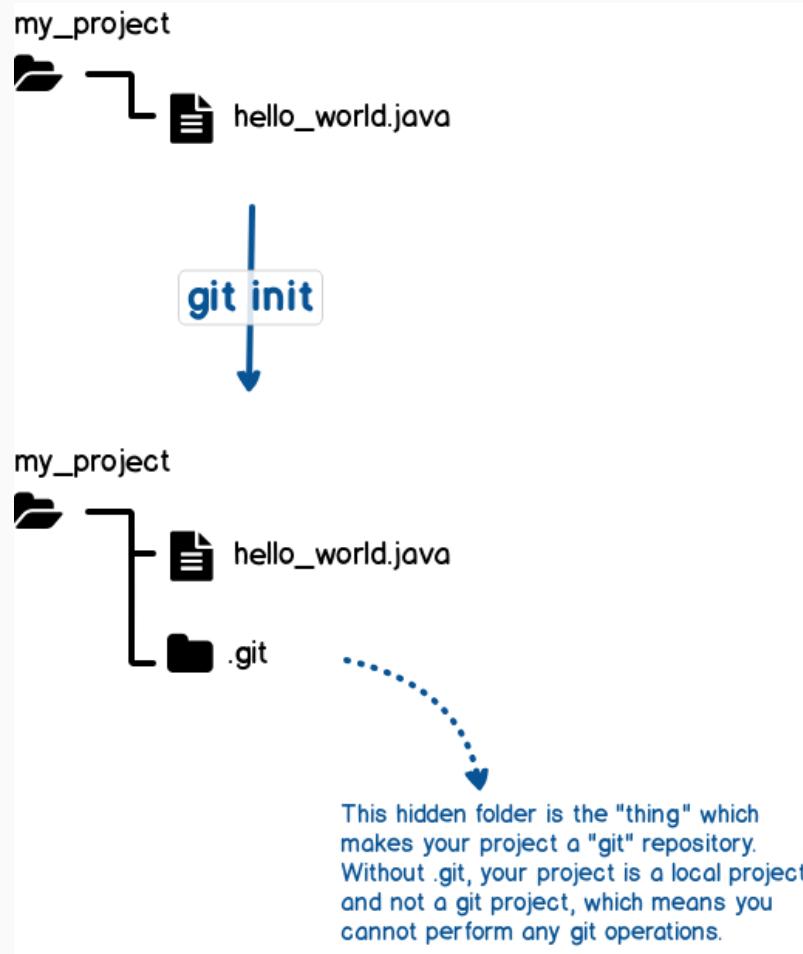
Forken o repositório da disciplina para o GitHub de vocês

<https://github.com/mauriciovancine/disciplina-analise-geoespacial-r>



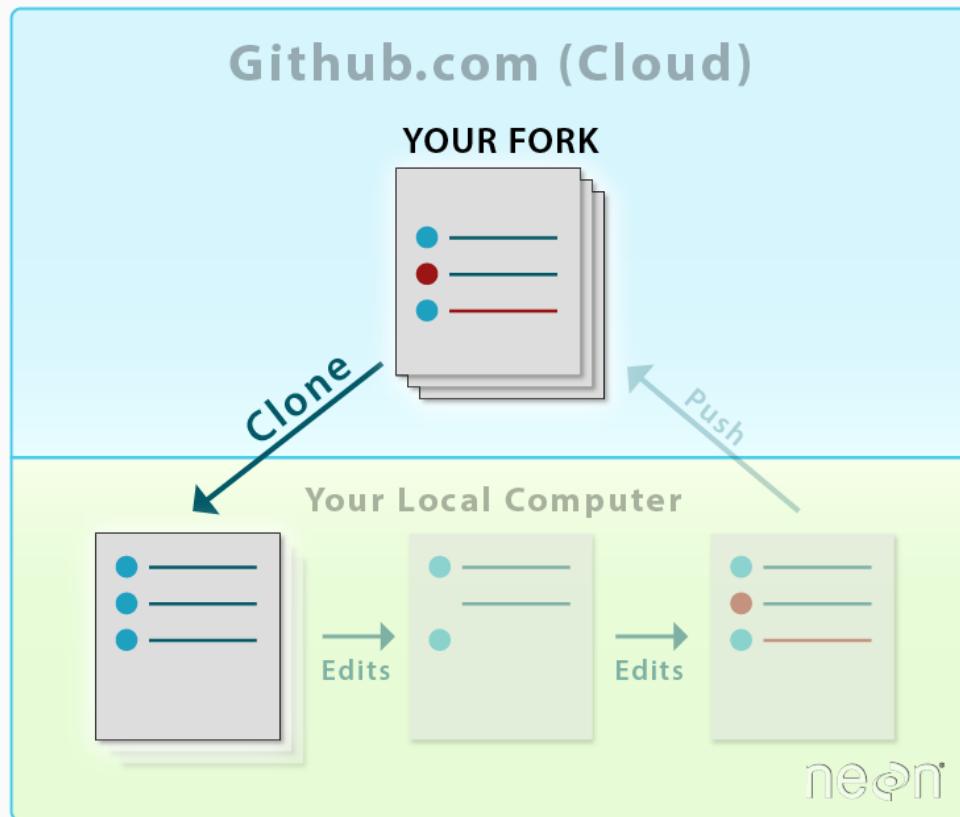
1.7 Iniciando: clone ou init

init: inicia o versionamento de um repositório



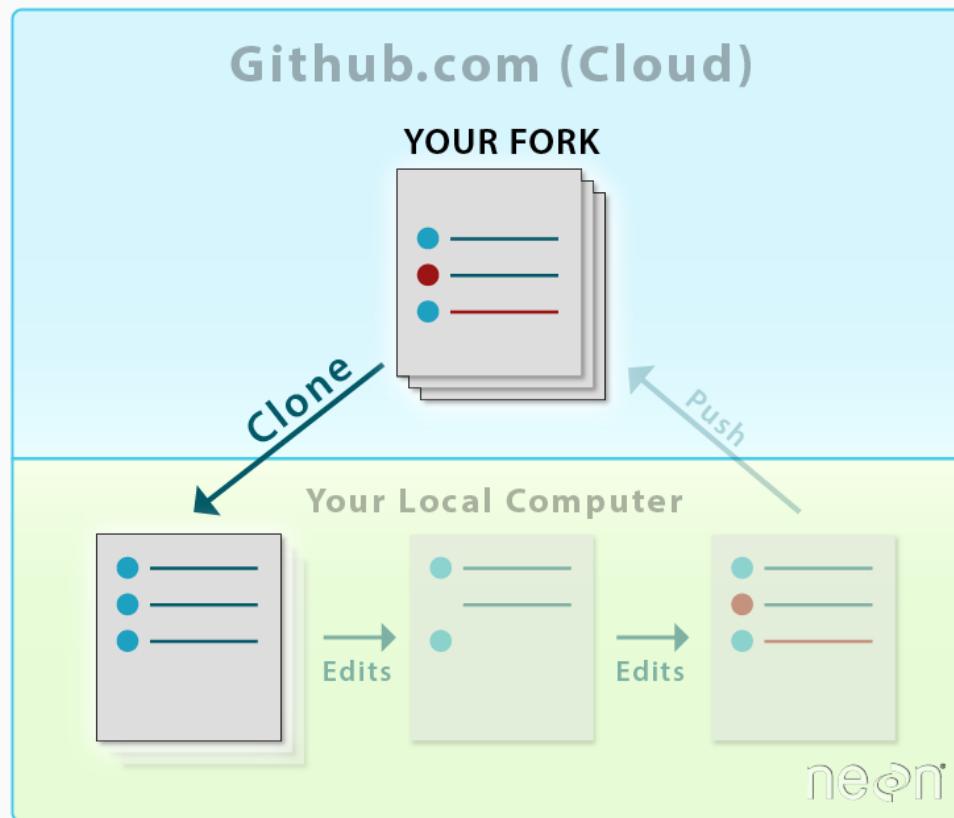
1.7 Iniciando: clone ou init

clone: copia um repositório remoto para o seu PC



1.7 Iniciando: clone ou init

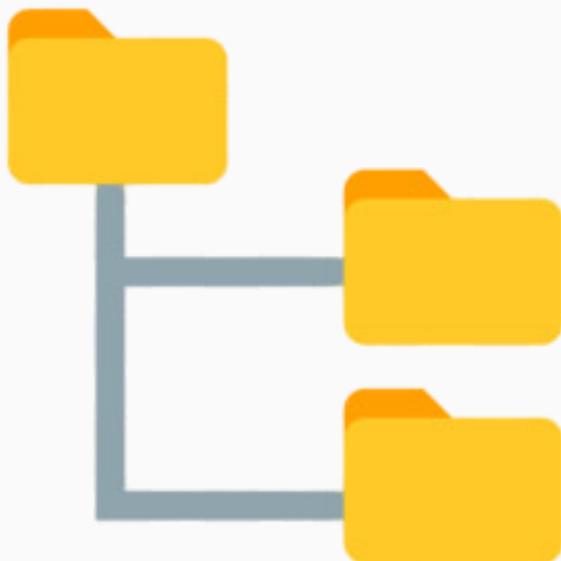
Vamos clonar o **repositório remoto (GitHub)** para o **repositório local (PC)**



1.7 Iniciando: clone ou init

Criem uma pasta chamada `github` no PC de vocês

```
".
"    ┌── home/
"        ┌── data/
"            ┌── github/          # todos os repositórios"
```



E como faremos isso?

1.7 Iniciando: clone ou init

1. Download diretamente do repositório remoto no formato .zip

The screenshot shows a GitHub repository page for 'mauriciovancine / disciplina-analise-geoespacial-r'. The 'Code' dropdown menu is highlighted with a red box. Below it, the 'Download ZIP' button is also highlighted with a red box. The repository has 1 branch and 0 tags. The commit history shows several files added on 2020-10-07, including '00_plano_ensino', '01_aulas', '02_scripts', '03_dados', '04_exercicios', '.gitignore', 'README.md', and 'disciplina-geoprocessamento.Rproj'. The 'About' section indicates no description, website, or topics provided. The 'Releases' section shows no releases published, with a link to 'Create a new release'. The 'Packages' section shows no packages published, with a link to 'Publish your first package'.

mauriciovancine / **disciplina-analise-geoespacial-r**

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Clone HTTPS SSH GitHub CLI
https://github.com/mauriciovancine/

Use Git or checkout with SVN using the web URL.

Download ZIP

About

No description, website, or topics provided.

Readme

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

1.7 Iniciando: clone ou init

2. Usar o terminal para clonar pelo HTTPS do repositório

The screenshot shows a GitHub repository page for 'mauriciovancine / disciplina-analise-geoespacial-r'. The 'Code' dropdown menu is open, highlighting the 'Clone' section which contains the HTTPS URL: <https://github.com/mauriciovancine/>. The repository details show 1 branch and 0 tags. The commit history lists several files and folders added on 2020-10-07, including '00_plano_ensino', '01_aulas', '02_scripts', '03_dados', '04_exercicios', '.gitignore', 'README.md', and 'disciplina-geoprocessamento.Rproj'. The repository has 1 unwatcher, 0 stars, and 1 fork.

1.7 Iniciando: clone ou init

3. Usar o RStudio para clonar pelo HTTPS do repositório



Vamos abrir o RStudio

RStudio

1

2

3

4

The screenshot shows the RStudio interface with four large black numbers overlaid on specific components:

- 1** is positioned over the Script Editor window, which contains R code for loading a dataset and creating a dygraph plot.
- 2** is positioned over the Console window, which displays the R session history, including the loading of the dygraphs library and the nhtemp dataset.
- 3** is positioned over the Environment window, showing the global environment with the nhtemp variable defined as a time series from 1912 to 1971.
- 4** is positioned over the Plots window, which displays a line graph titled "New Haven Temperatures" showing temperature fluctuations over time, with a corresponding dyRangeSelector at the bottom.

Code in the Script Editor (Section 1):

```
library(dygraphs)
data("nhtemp")
nhtemp
dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyRangeSelector(dateWindow = c("1920-01-01", "1960-01-01"))
```

Console History (Section 2):

```
> library(dygraphs)
> data("nhtemp")
> 
nhtemp
Time Series:
Start = 1912
End = 1971
Frequency = 1
[1] 49.9 52.3 49.4 51.1 49.4 47.9 49.8 50.9 49.3 51.9 50.8 49.6 49.3 50.6 48.4 50.7 50.9 50.6 51.5
[20] 52.8 51.8 51.1 49.8 50.2 50.4 51.6 51.8 50.9 48.8 51.7 51.0 50.6 51.7 51.5 52.1 51.3 51.0 54.0
[39] 51.4 52.7 53.1 54.6 52.0 52.0 50.9 52.6 50.2 52.6 51.6 51.9 50.5 50.9 51.7 51.4 51.7 50.8 51.9
[58] 51.8 51.9 53.0

> dygraph(nhtemp, main = "New Haven Temperatures") %>%
+   dyRangeSelector(dateWindow = c("1920-01-01", "1960-01-01"))
> |
```

Environment (Section 3):

Values

```
nhtemp Time-Series [1:60] from 1912 to 1971: 49.9 52.3 49.4 51.1 ...
```

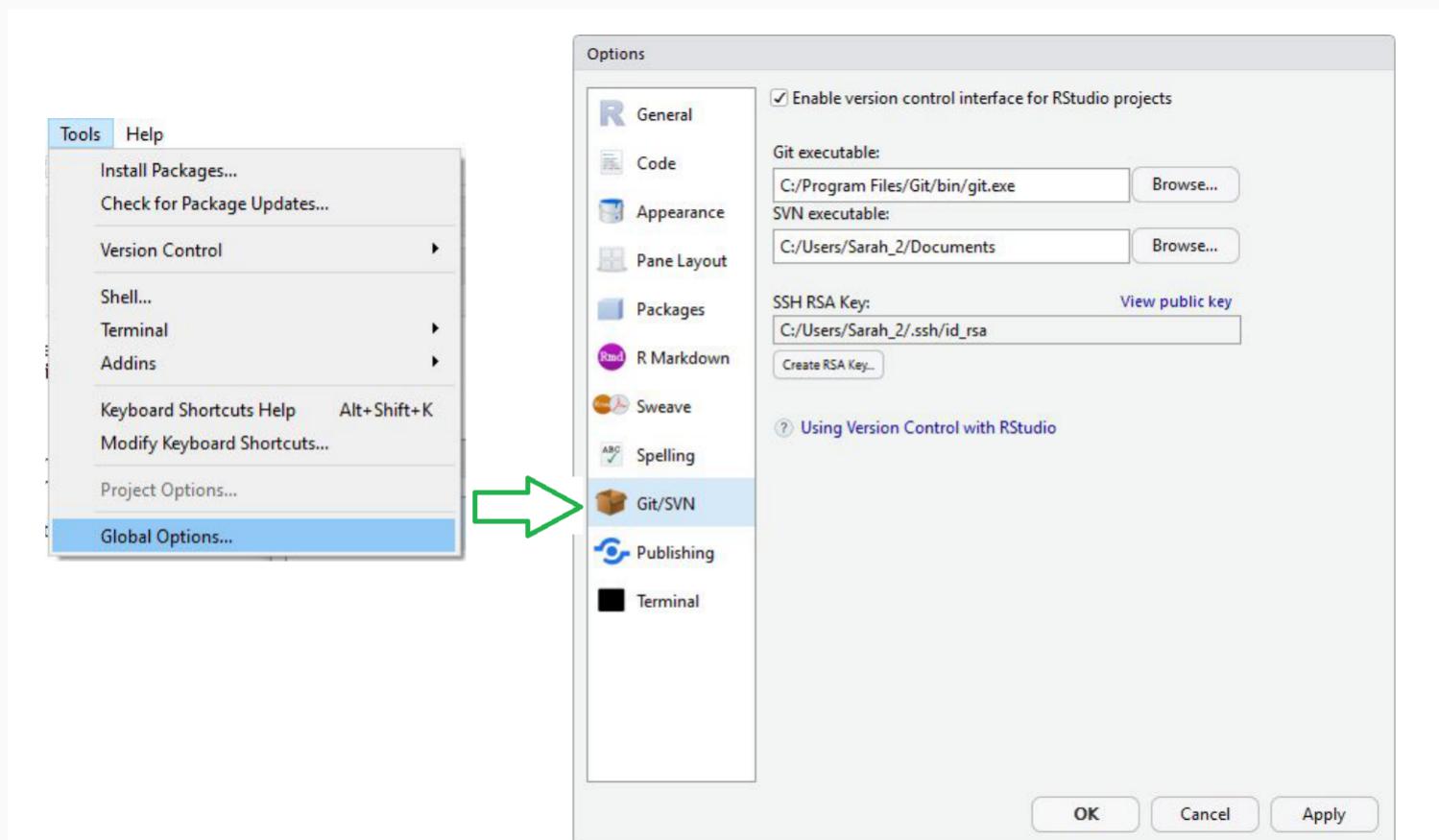
Plots (Section 4):

New Haven Temperatures

Git e RStudio

Indicar o caminho do git para o RStudio

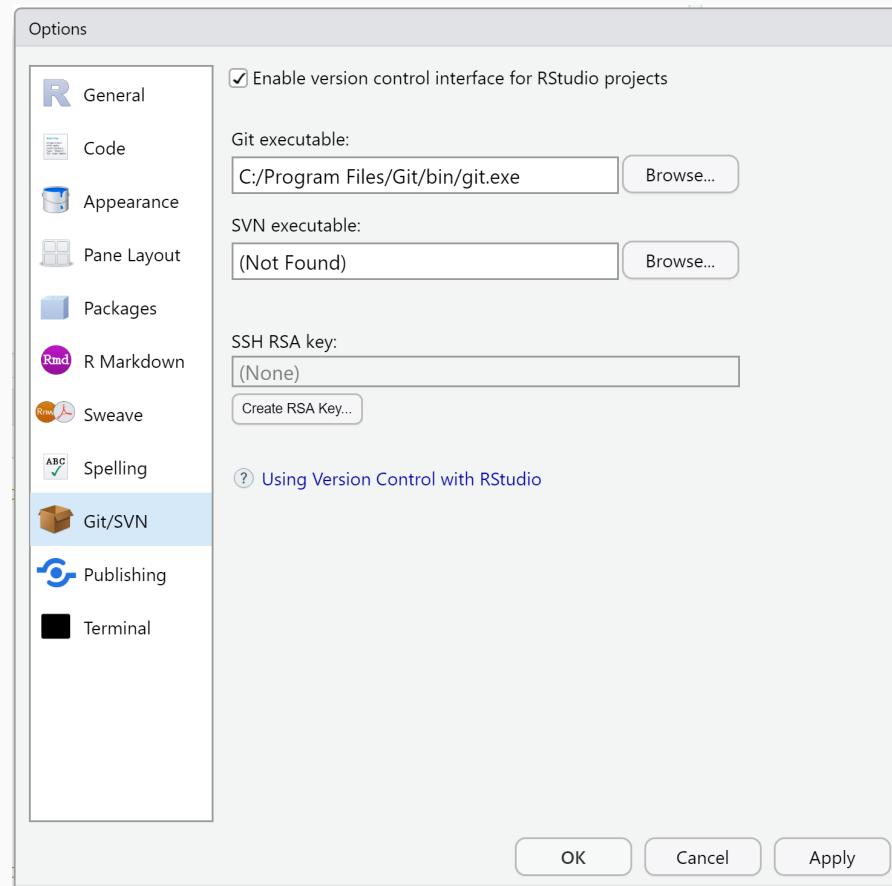
- Tools > Global Options > Git/SVN



Git e RStudio

Marcar Enable version... e Git executable:

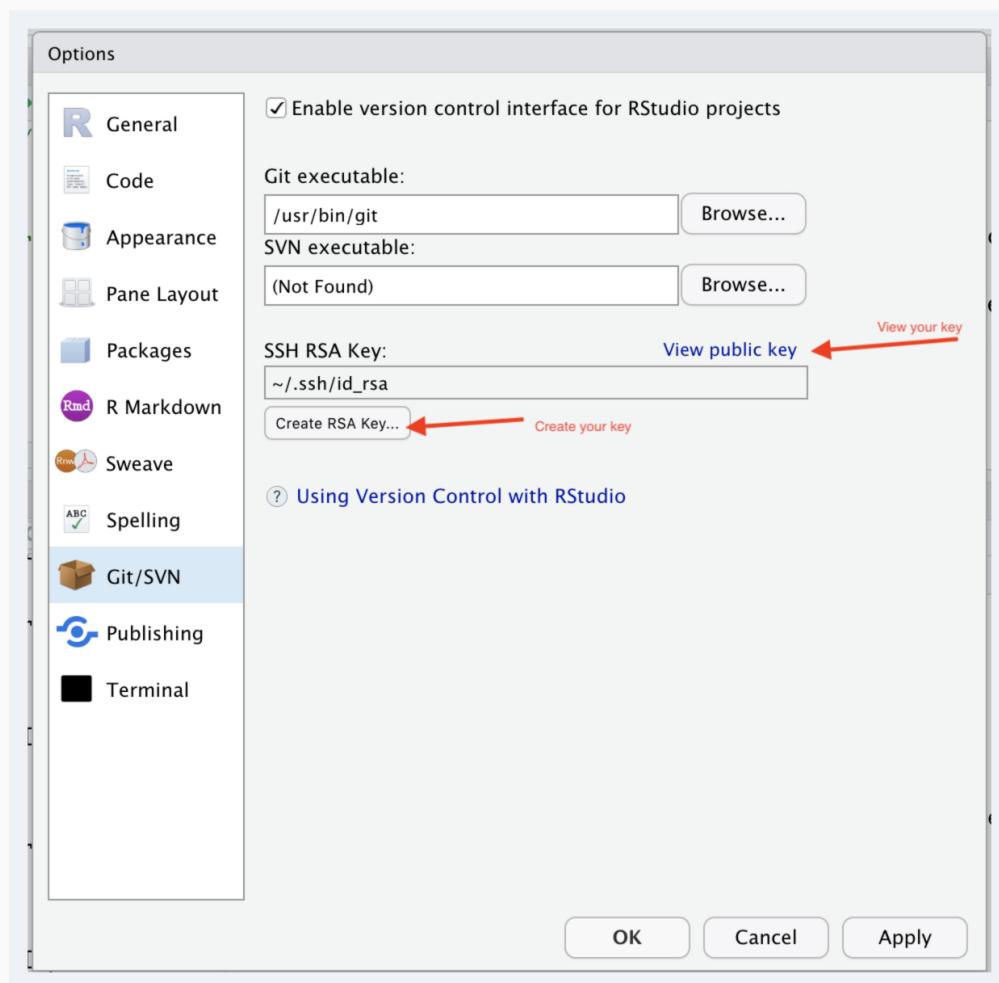
- Windows: C:/Program Files/Git/bin/git.exe
- Ubuntu: /usr/bin/git



Git e RStudio

Criar uma chave SSH

- Create RSA key
- View public key
- Copiar: Ctrl + C



GitHub e RStudio

No GitHub...

- Perfil > Settings > SSH and GPG keys > New SSH key

The screenshot shows the GitHub user interface for managing SSH and GPG keys. On the left, there's a sidebar with links like Profile, Account, Account security, Security log, Security & analysis, Emails, Notifications, Scheduled reminders, Billing, and SSH and GPG keys (which is highlighted with a red box). The main content area has two sections: 'SSH keys' and 'GPG keys'. The 'SSH keys' section displays a single key entry for 'ubuntu_tor' with details: SHA256 fingerprint, date added (May 3, 2020), and usage information ('Last used within the last 2 weeks — Read/write'). It includes a 'Delete' button and a 'New SSH key' button (also highlighted with a red box). The 'GPG keys' section indicates 'There are no GPG keys associated with your account' and provides a link to generate one.

mauriciovancine Personal settings

Profile

Account

Account security

Security log

Security & analysis

Emails

Notifications

Scheduled reminders

Billing

SSH and GPG keys

Repositories

Organizations

SSH keys

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

	ubuntu_tor	SHA256:30g9YxKAndQVqnITS49x79UfFCXu7XT/1x30kNQr/rc	Delete
Added on May 3, 2020			
Last used within the last 2 weeks — Read/write			

New SSH key

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key](#) and add it to your account.

GitHub e RStudio

No GitHub...

- Title: um nome qualquer
- Key: colar (Ctrl + V)
- Add SSH key

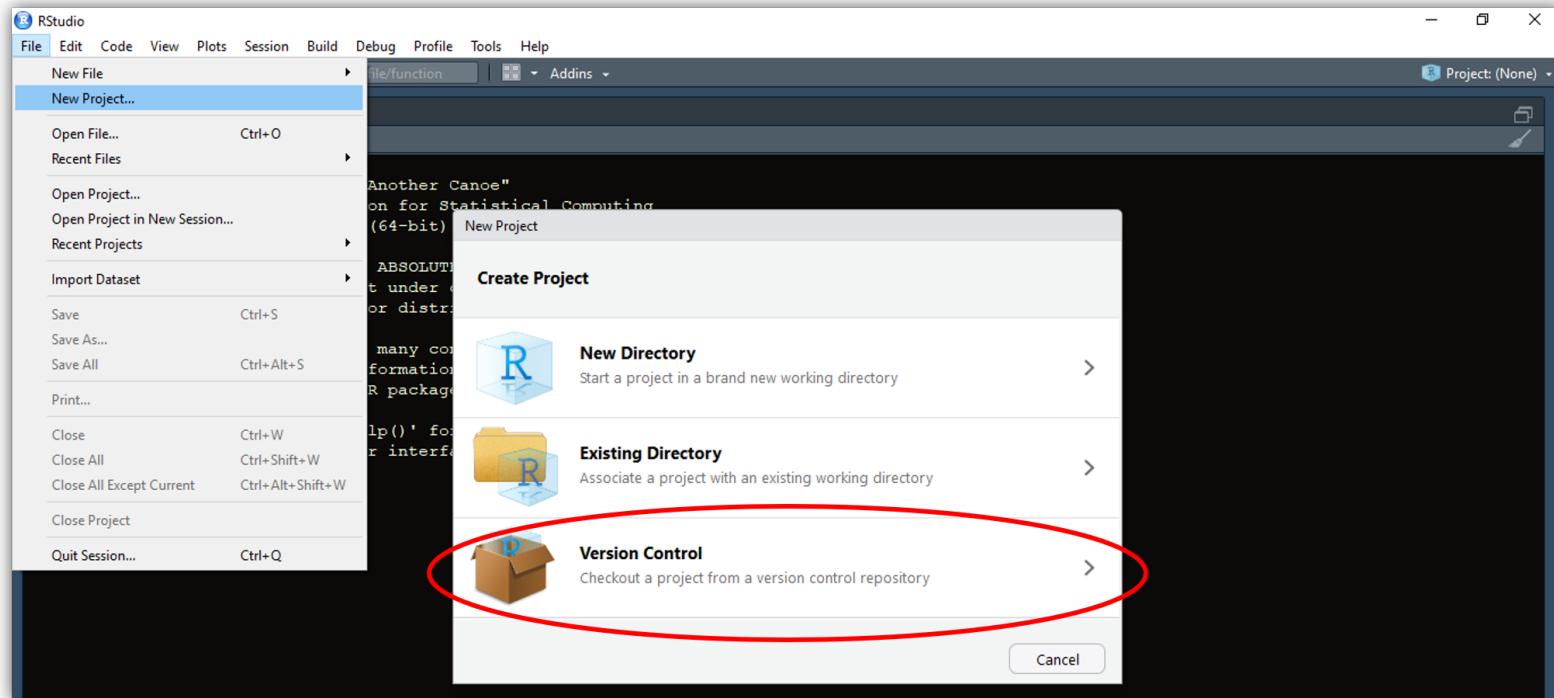
The screenshot shows the GitHub 'Personal settings' sidebar on the left with the 'SSH and GPG keys' option selected. The main area is titled 'SSH keys / Add new'. It contains two input fields: 'Title' (with a placeholder 'Title') and 'Key' (with a placeholder 'Begins with 'ssh-rsa', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521''). A green 'Add SSH key' button is at the bottom. Red boxes highlight the 'Title' input field, the 'Key' text area, and the 'Add SSH key' button.

Tudo certo até aqui?

1.7 Iniciando: clone ou init

Criar um Projeto R com controle de versão

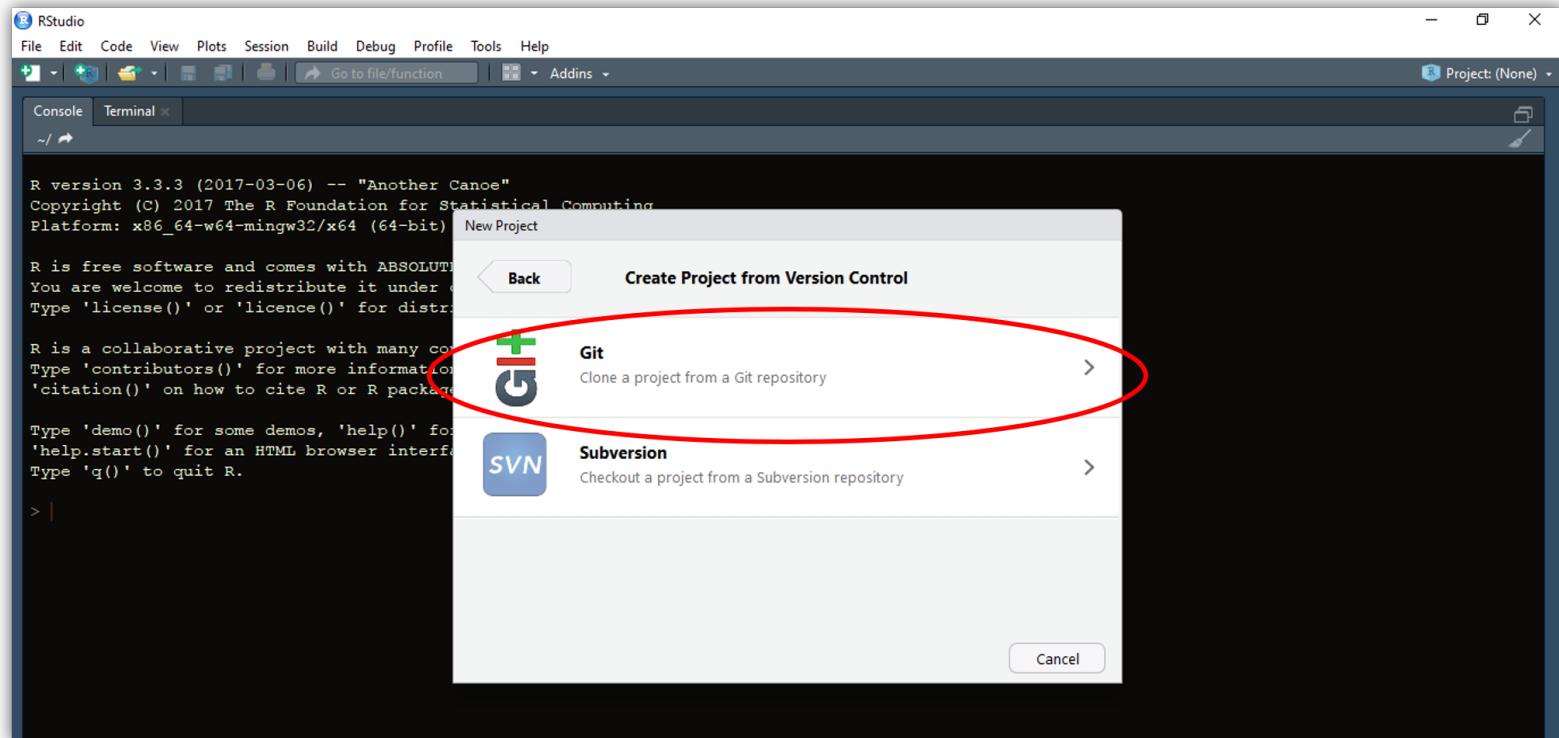
File > New Project > Version Control



1.7 Iniciando: clone ou init

Escolher clonar repositório do GitHub

Git

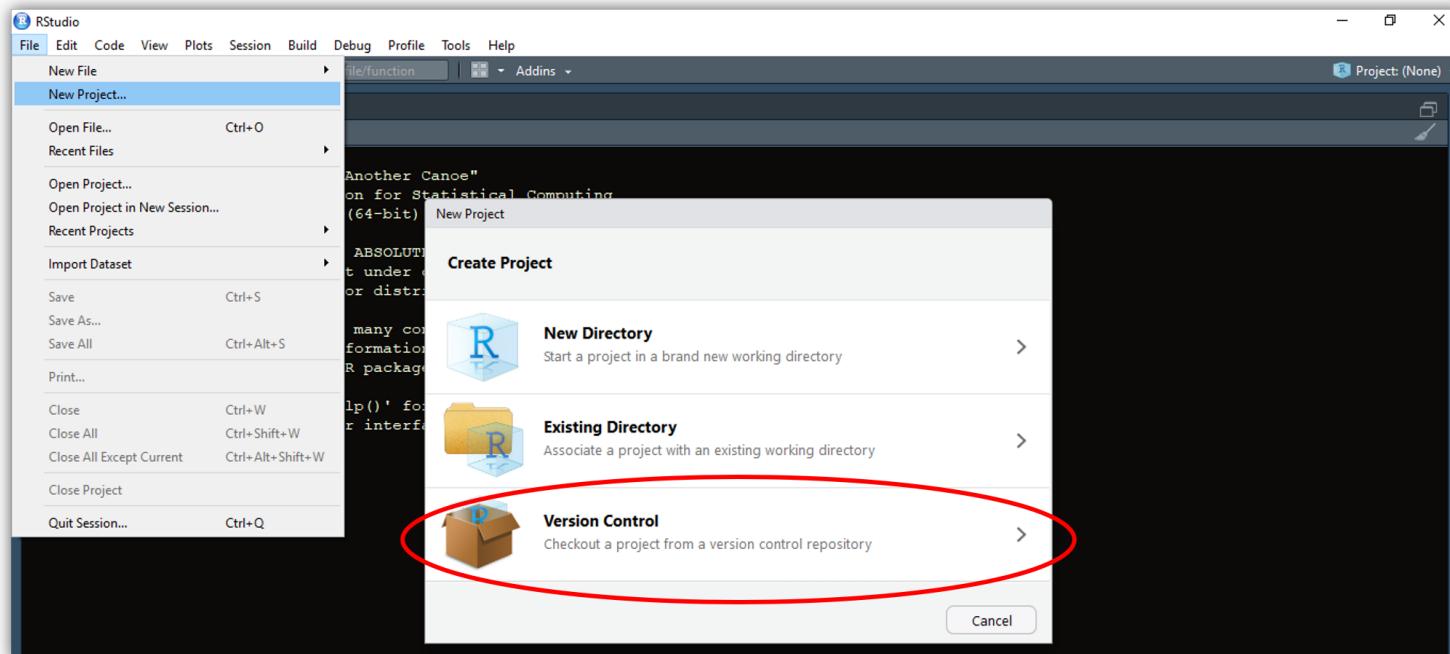


1.7 Iniciando: clone ou init

Preencher:

Repository URL: <https://github.com/mauriciovancine/disciplina-analise-geoespacial-r.git>

Create project as subdirectory of: /home/mude/data/github



1.7 Iniciando: clone ou init

Terminal

Na aba **Terminal** do RStudio, digite:

```
# terminal  
git clone https://github.com/mauriciovancine/disciplina-analise-geoespacial-
```



1.8 Configurando: config

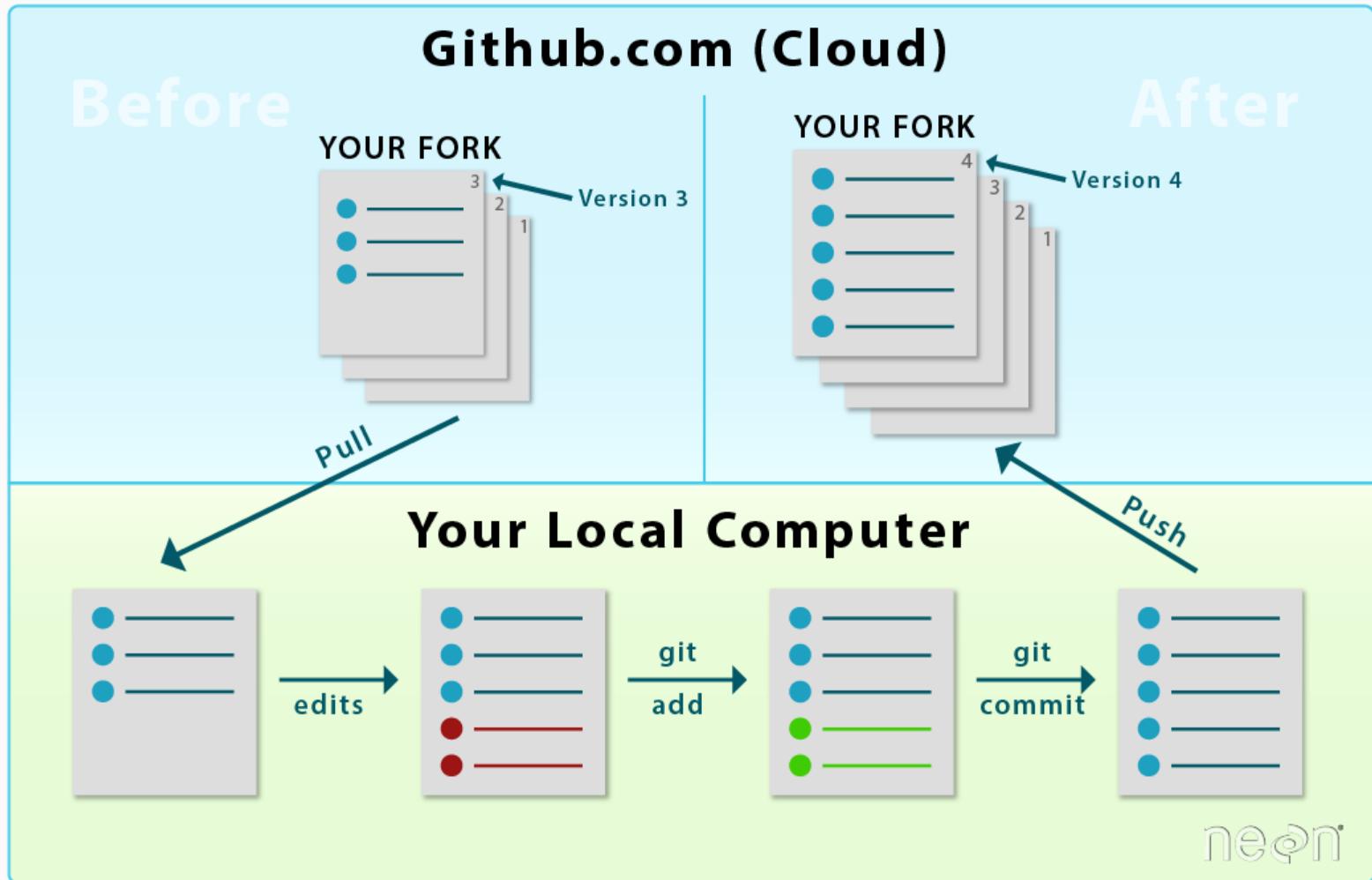
Precisamos **definir as configurações** do git

```
# terminal  
git config --list  
git config --global user.name "meu nome"  
git config --global user.email "email@dominio.com"  
git config --list
```



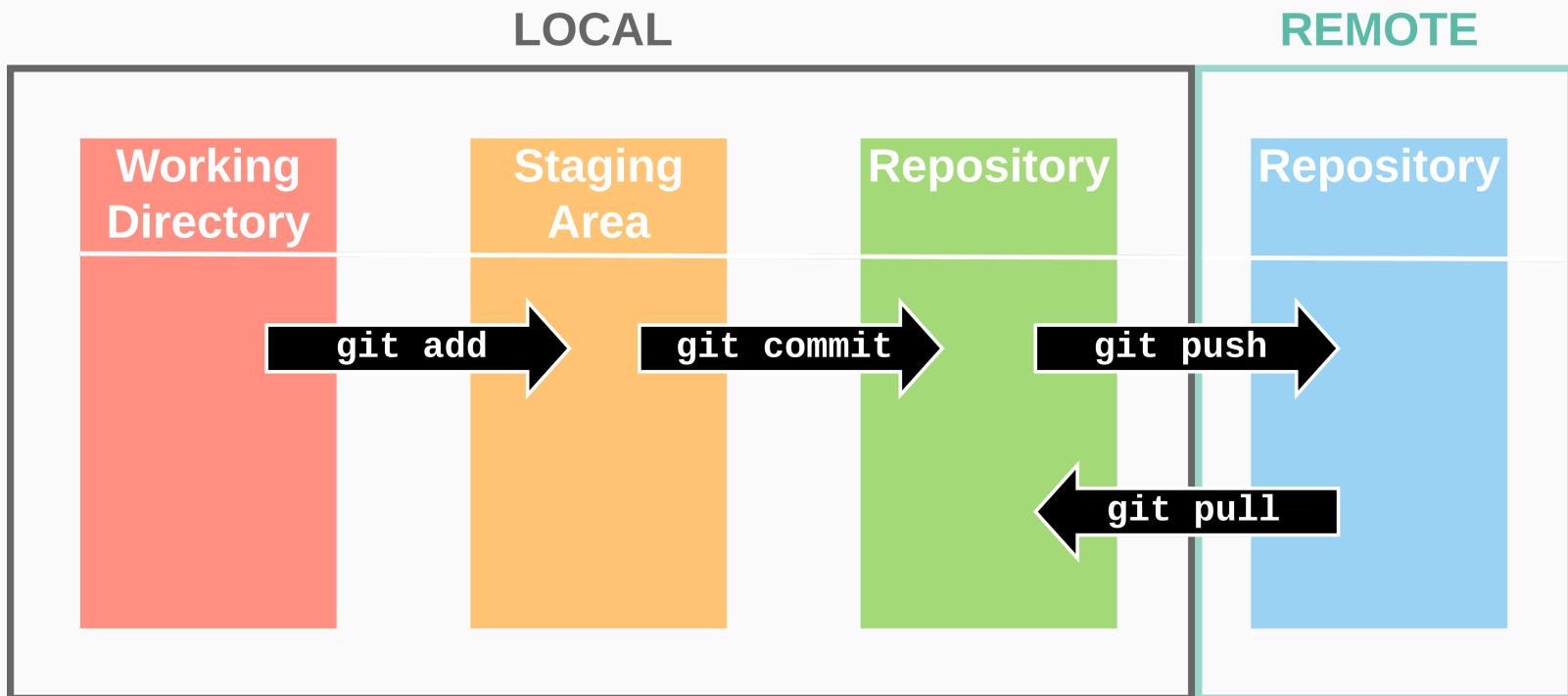
Git Config

Fluxo de trabalho com o git e GitHub



Fluxo de trabalho com o git e GitHub

Comandos para usar o git

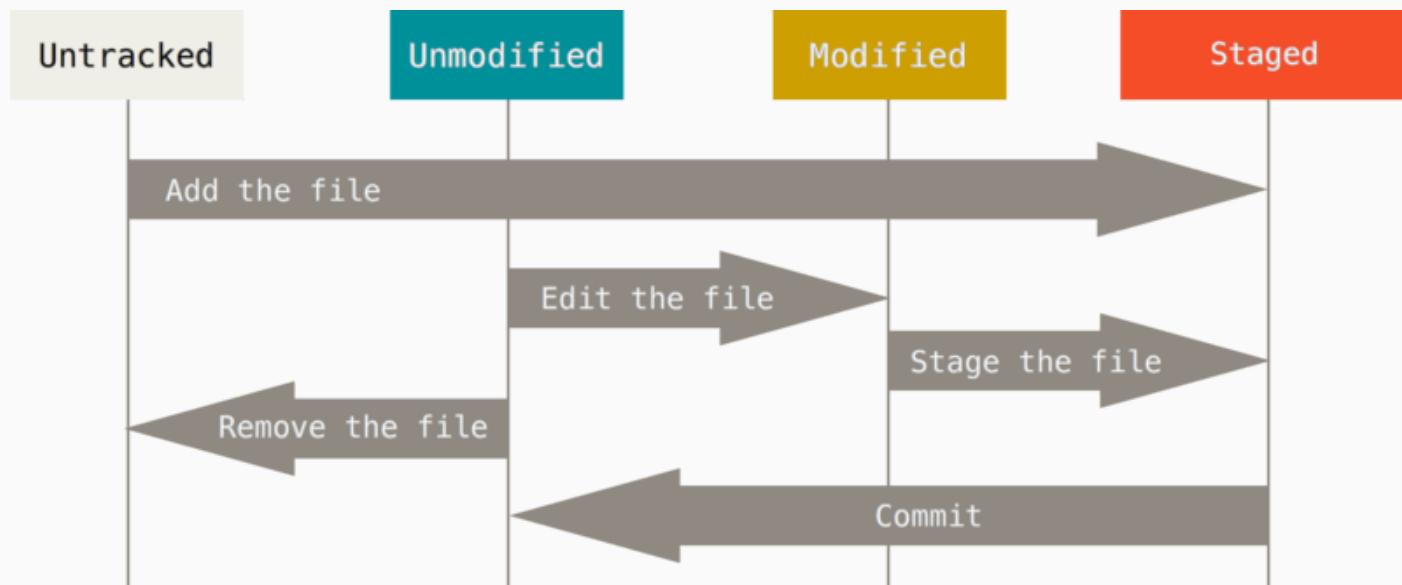


Agora prestem atenção porque complica... =]

Git Status

Estados do repositório local do git:

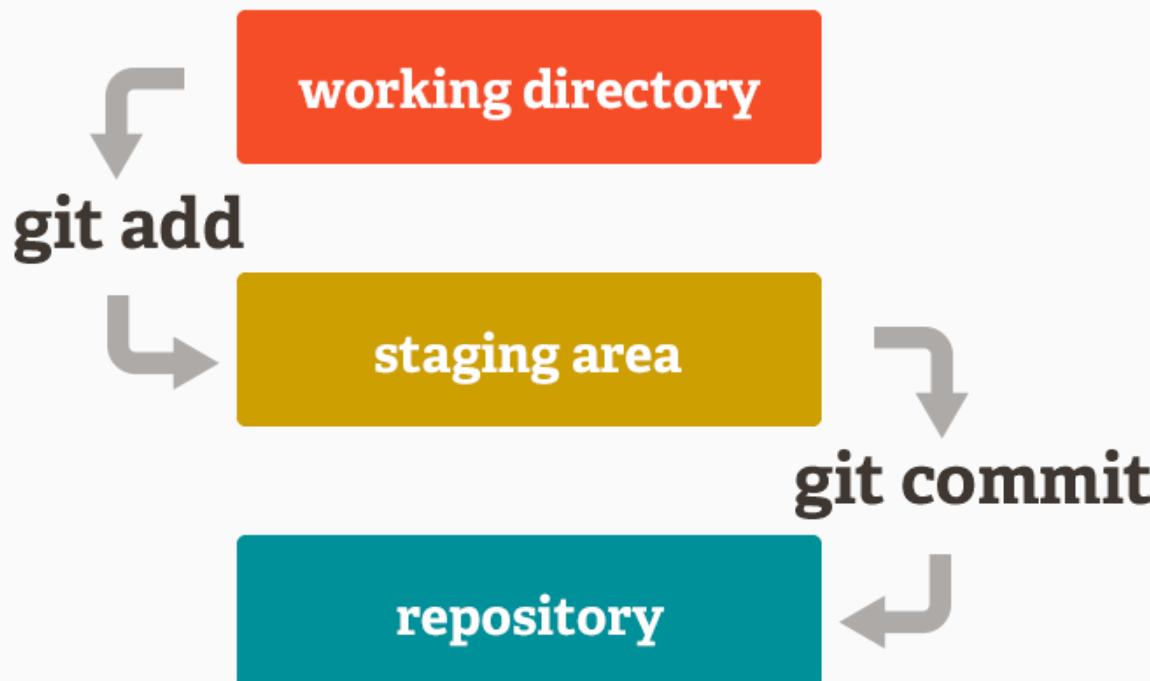
1. **Untracked**: não registrou o arquivo criado ou deletado
 2. **Unmodified**: marcou os arquivos como não-modificados após o commit
 3. **Modified**: marcou os arquivos modificados depois da edição
 4. **Staged**: marcou o arquivo adicionado ou modificado para ir para o commit
- **Commit**: os arquivos são armazenados no banco de dados local (.git)



Git Sections

Seções de um repositório git:

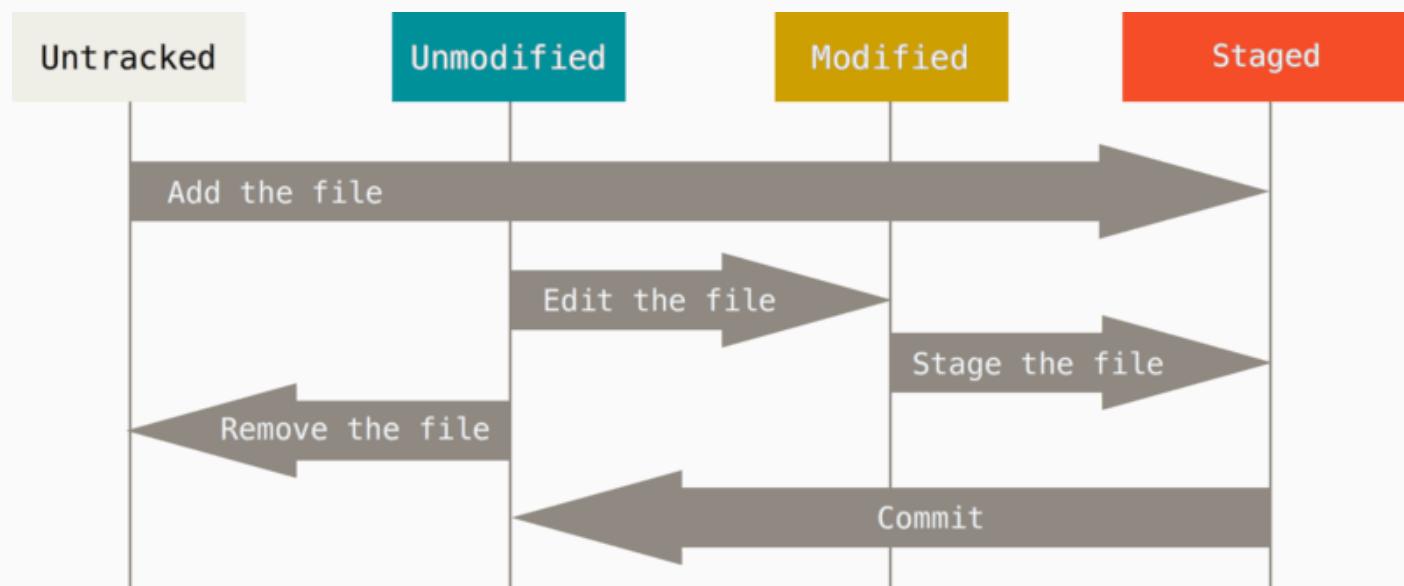
1. **Working directory**: arquivos não-trakeados
2. **Staging area**: arquivos trackeados (adicionados ou modificados)
3. **Repository**: arquivos comitados (.git)



1.9 Básico: status, add, commit, e log

status: mostra o status do repositório e dos arquivos

```
git status
```



Vamos criar um arquivo no diretório

Usando a aba **Terminal** do RStudio

```
# terminal  
touch teste.txt
```

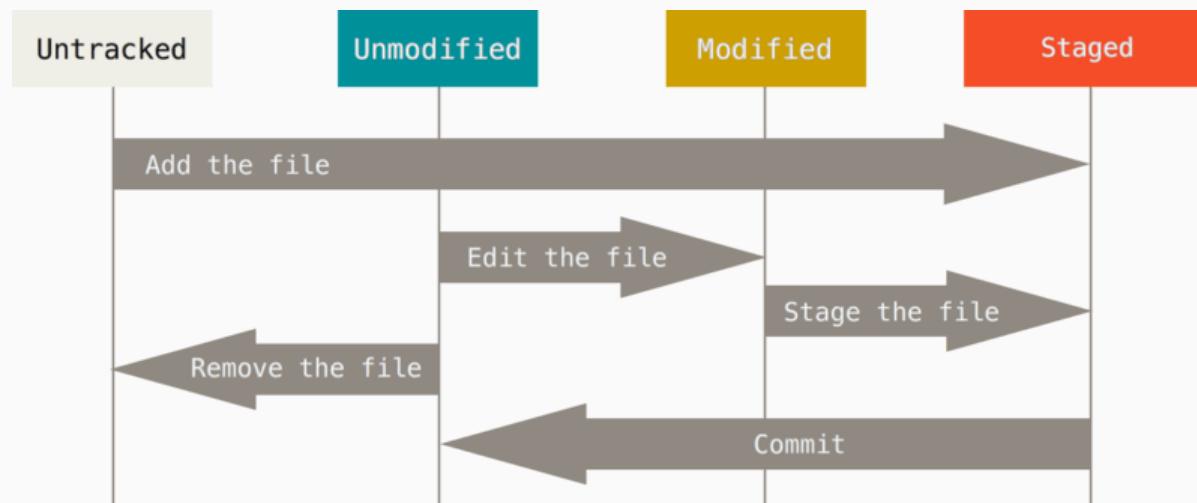


1.9 Básico: status, add, commit, e log

status: mostra o status do repositório e dos arquivos

```
git status
```

```
## Untracked files:  
##   (use "git add <file>..." to include in what will be committed)  
##       ./teste.txt  
##  
## no changes added to commit (use "git add" and/or "git commit -a")
```

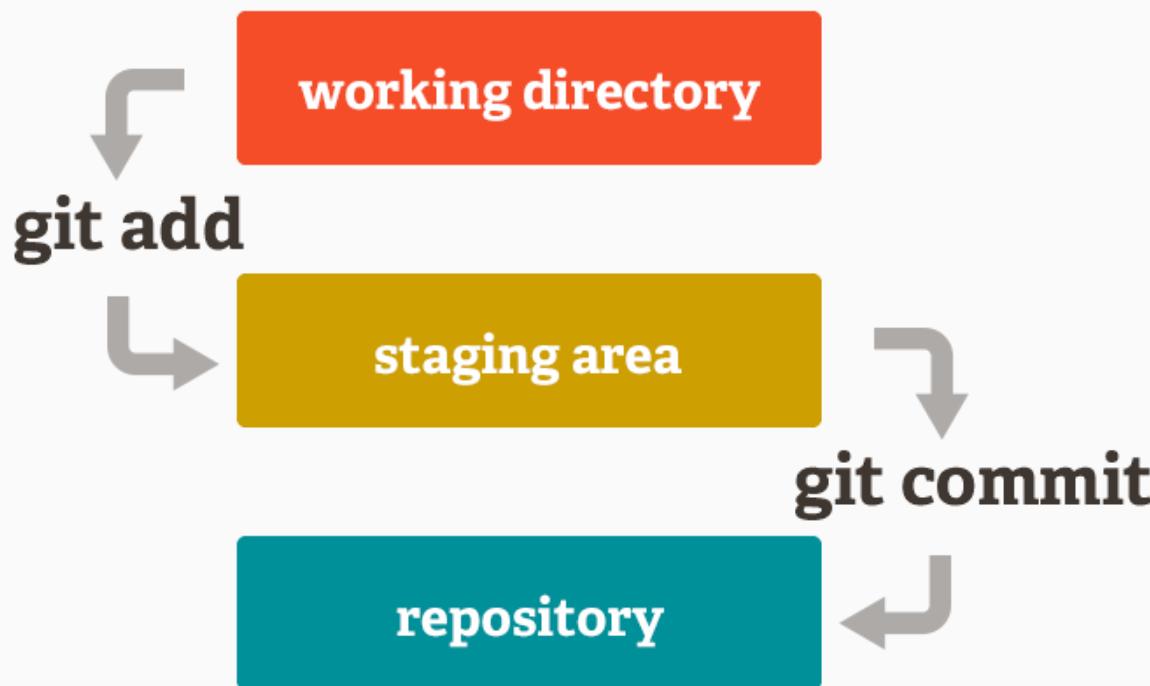


1.9 Básico: status, add, commit, e log

add: adiciona mudanças após edições (staging area)

```
# terminal  
git add -Av
```

```
## add 'teste.txt'
```

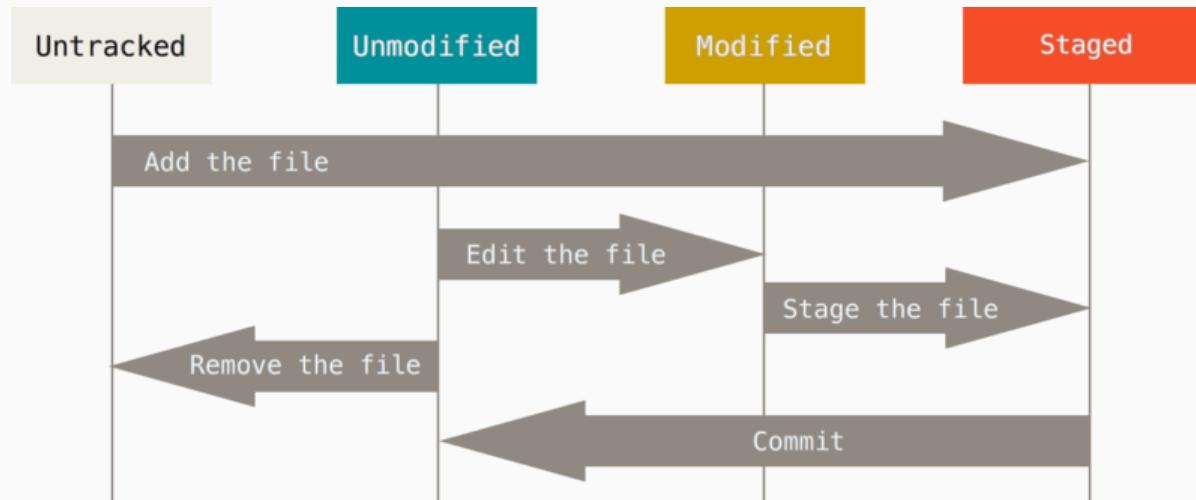


1.9 Básico: status, add, commit, e log

status: mostra o status do repositório e dos arquivos

```
git status
```

```
## Changes to be committed:  
##   (use "git restore --staged <file>..." to unstage)  
##       new file:  teste.txt
```



Vamos editar o arquivo no diretório

Usando a aba **Files** do RStudio abra e edite o arquivo **teste.txt**, inserindo:

```
Acabo de inserir um edição ao meu arquivo teste.txt
```

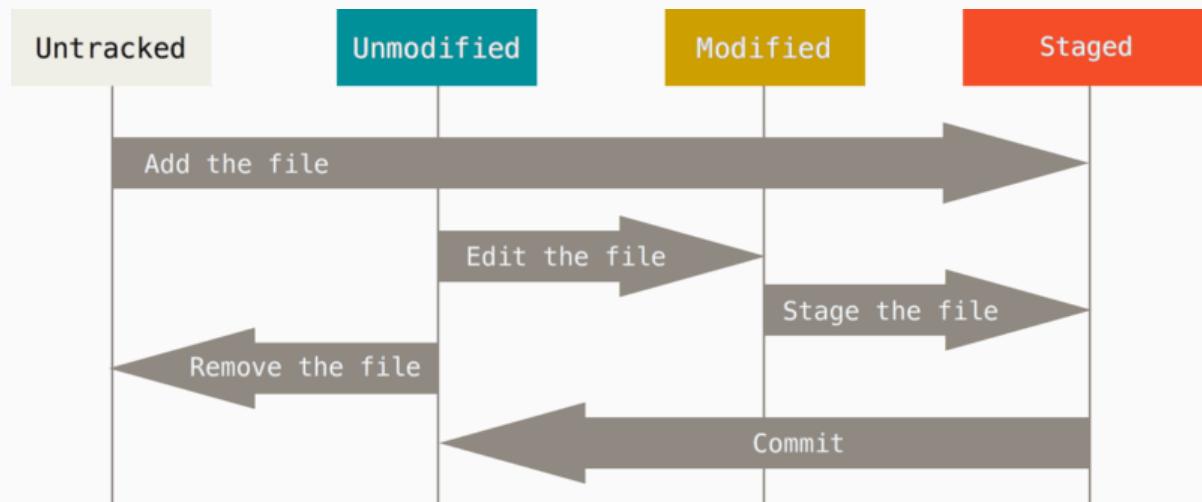


1.9 Básico: status, add, commit, e log

status: mostra o status do repositório e dos arquivos

```
git status
```

```
## Changes not staged for commit:  
##   (use "git add <file>..." to update what will be committed)  
##   (use "git restore <file>..." to discard changes in working directory)  
##       modified:   teste.txt
```

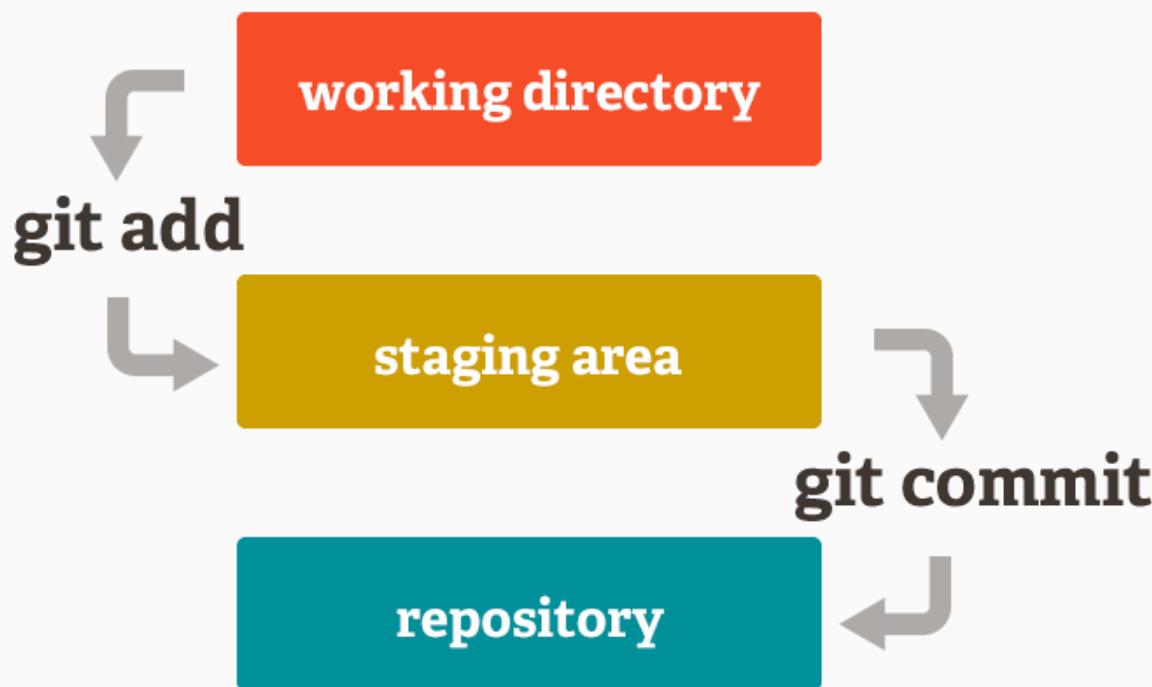


1.9 Básico: status, add, commit, e log

add: adiciona mudanças após edições (staging area)

```
# terminal  
git add -Av
```

```
add 'teste.txt'
```

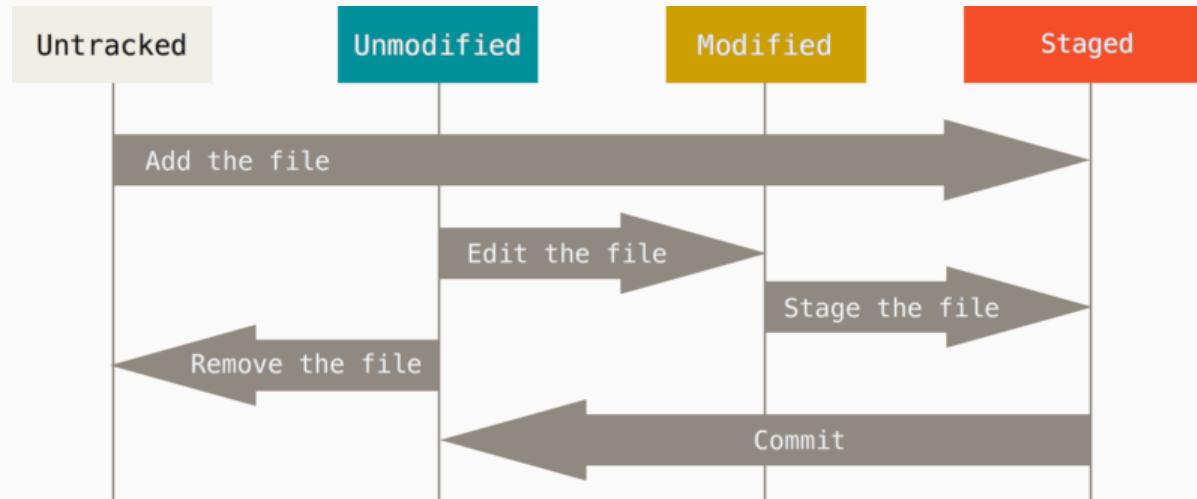


1.9 Básico: status, add, commit, e log

status: mostra o status do repositório e dos arquivos

```
git status
```

```
## Changes to be committed:  
##   (use "git restore --staged <file>..." to unstage)  
##       new file:  teste.txt
```

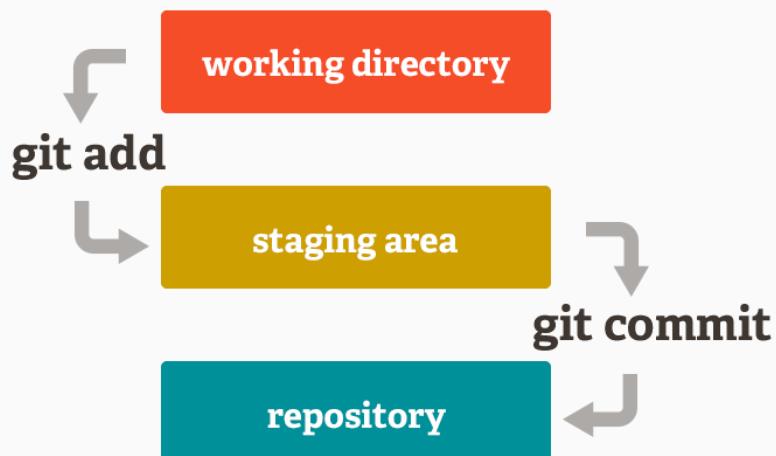


1.9 Básico: status, add, commit, e log

commit: armazena as mudanças com uma descrição , criando uma versão do repositório (repository)

```
git commit -m "2020-10-19 add teste.txt"
```

```
## [master 06211a1] 2020-10-19 add teste.txt
##   1 file changed, 1 insertion(+)
##   create mode 100644 teste.txt
```



1.9 Básico: status, add, commit, e log

ATENÇÃO!

	COMMENT	DATE
O	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
O	ENABLED CONFIG FILE PARSING	9 HOURS AGO
O	MISC BUGFIXES	5 HOURS AGO
O	CODE ADDITIONS/EDITS	4 HOURS AGO
O	MORE CODE	4 HOURS AGO
O	HERE HAVE CODE	4 HOURS AGO
O	AAAAAAA	3 HOURS AGO
O	ADKFJSLKDFJSDFKJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

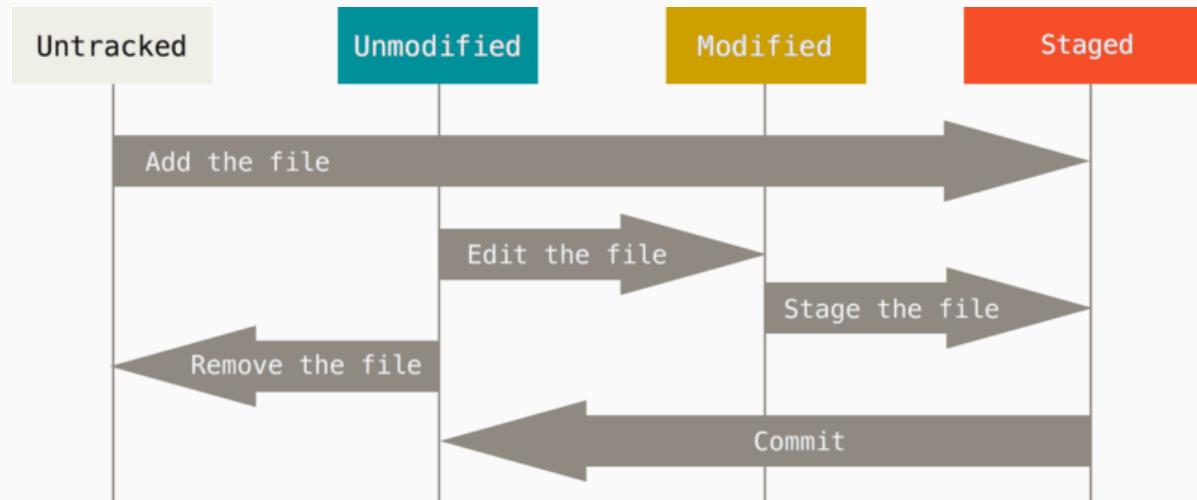
AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

1.9 Básico: status, add, commit, e log

status: mostra o estatus do repositório

```
git status
```

```
## Changes to be committed:  
##   (use "git restore --staged <file>..." to unstage)  
##       new file:  teste.txt
```

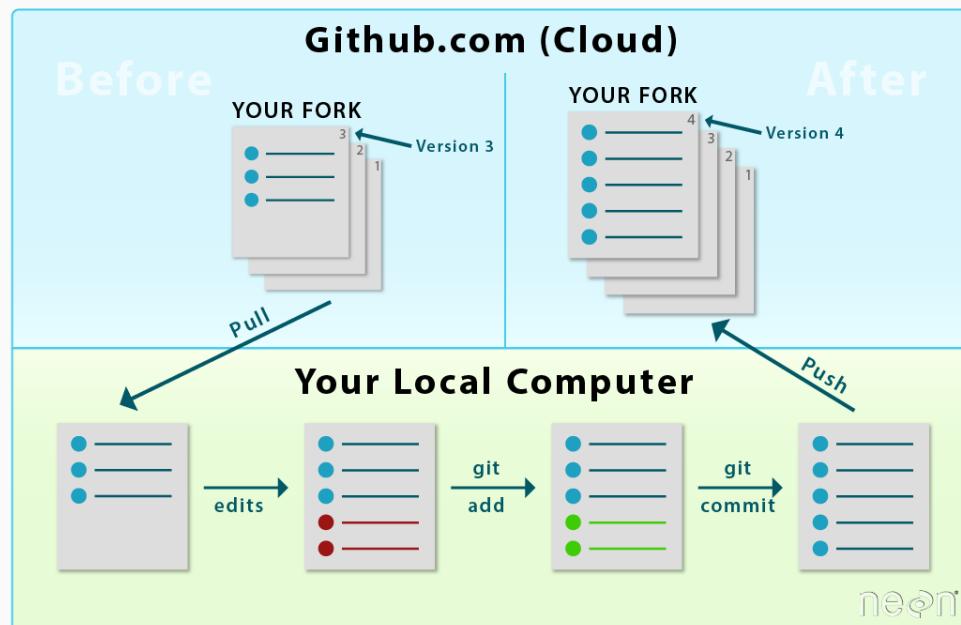


1.9 Básico: status, add, commit, e log

log: registro de todo o histórico de commits (tags)

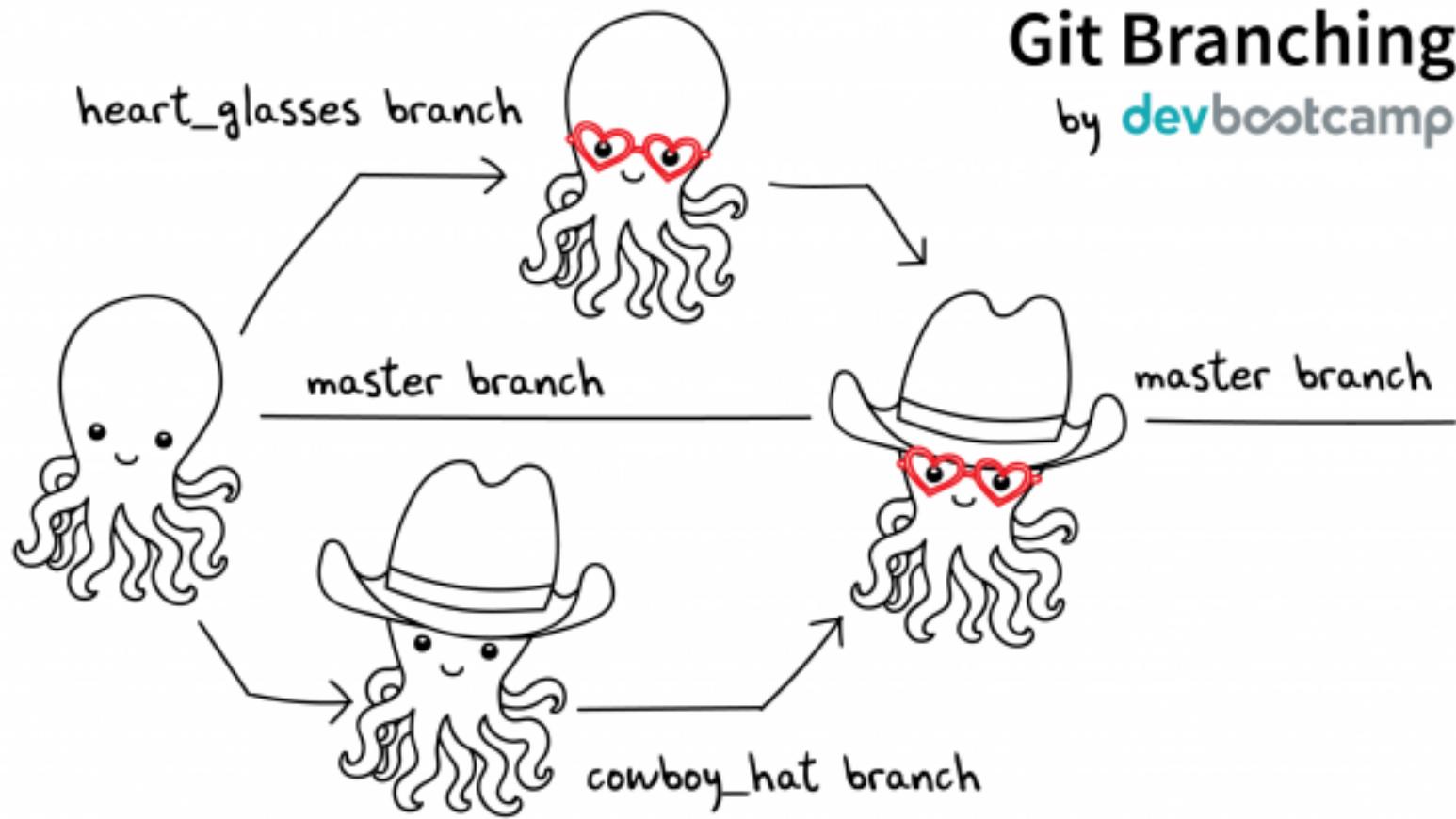
```
git log
```

```
## Changes to be committed:  
##   (use "git restore --staged <file>..." to unstage)  
##       new file:  teste.txt
```



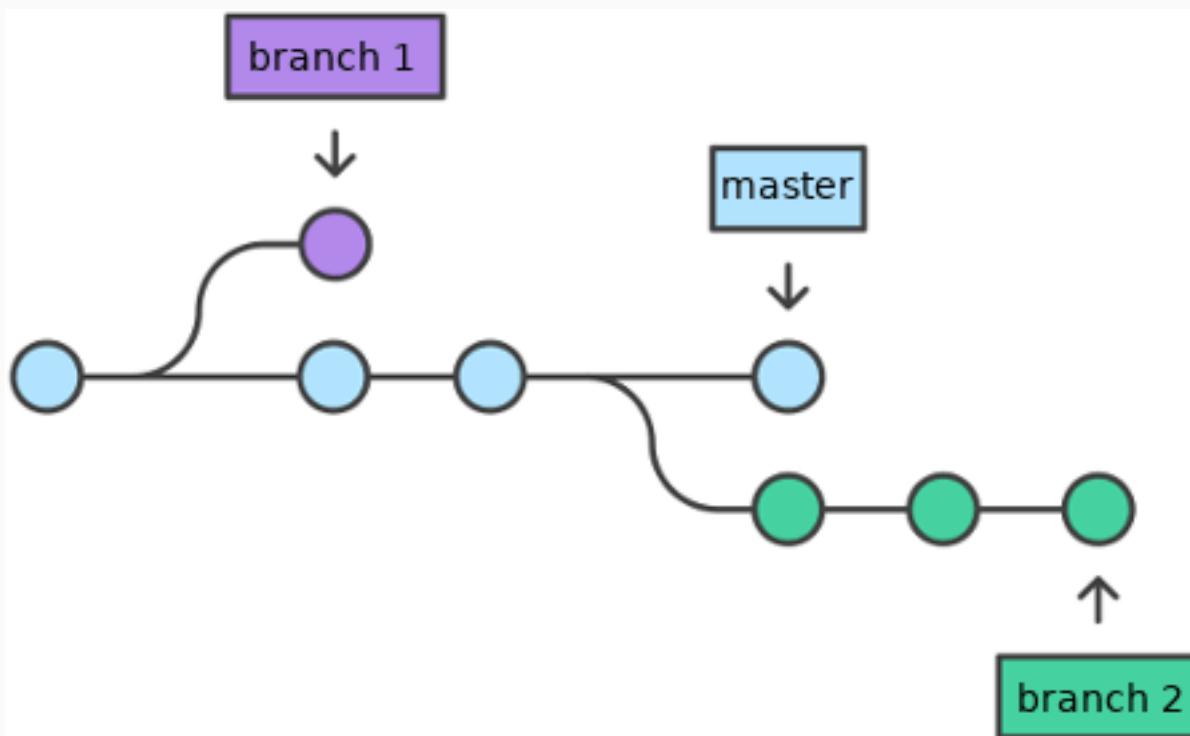
Git Branching

by **devbootcamp**



1.10 Ramificações: branch, switch e merge

branch: ramificação representa uma **linha independente** de desenvolvimento

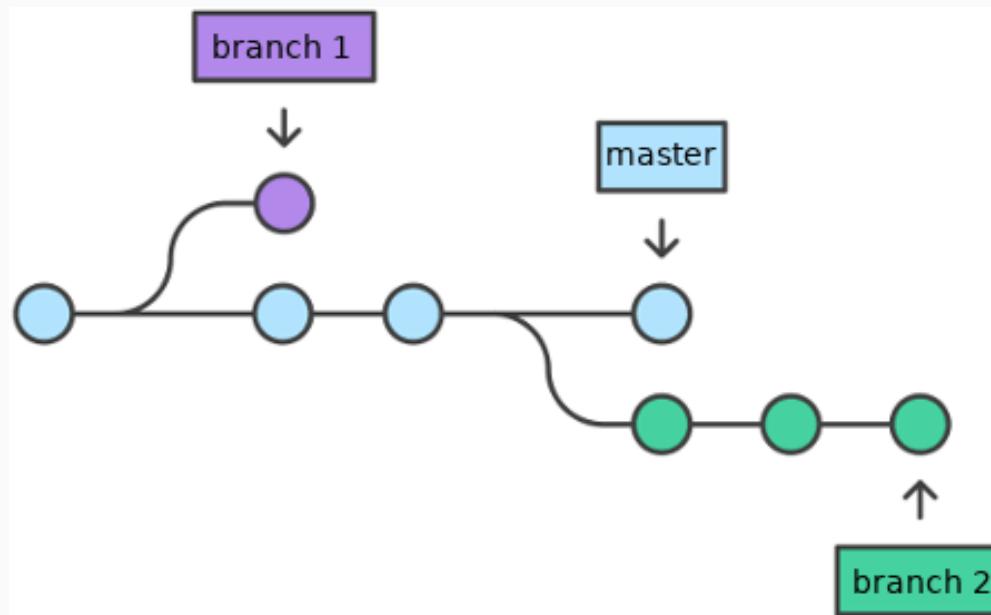


1.10 Ramificações: branch, switch e merge

branch: listar os branches e verificar o branch de edição

```
git branch
```

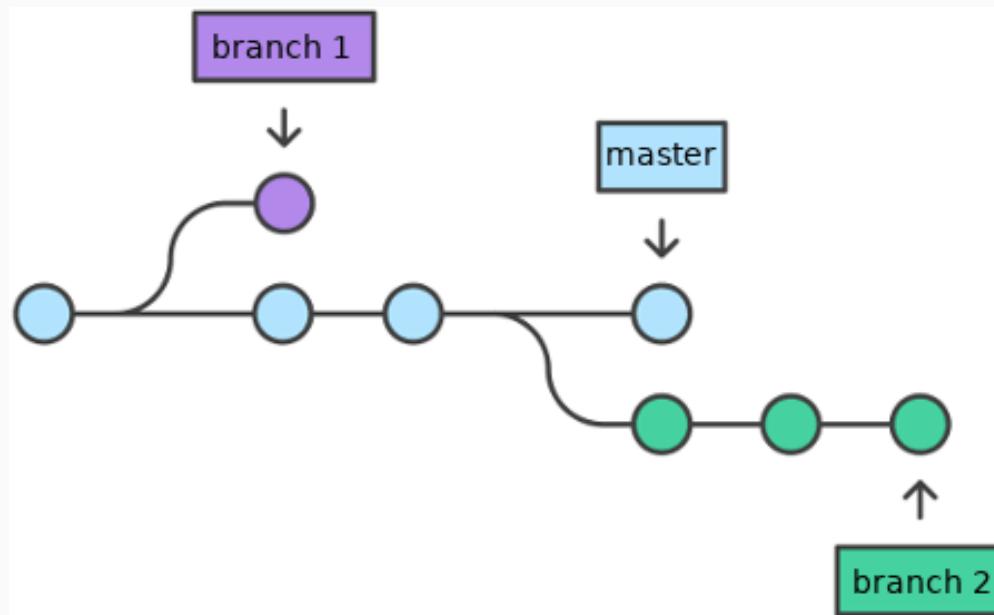
```
## * master
```



1.10 Ramificações: branch, switch e merge

branch: criar uma ramificação

```
git branch branch-1
```

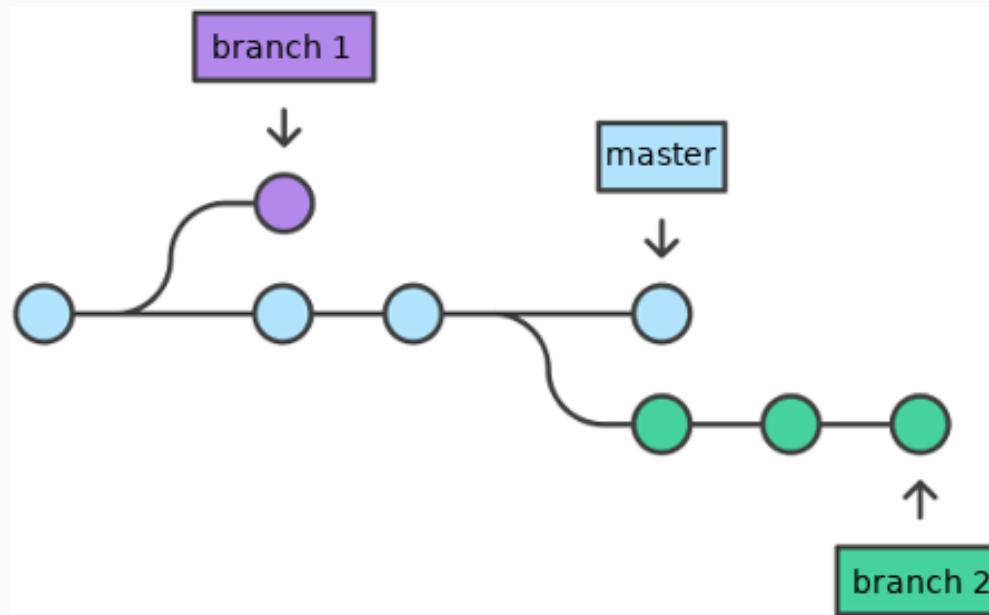


1.10 Ramificações: branch, switch e merge

branch: listar os branches e verificar o branch de edição

```
git branch
```

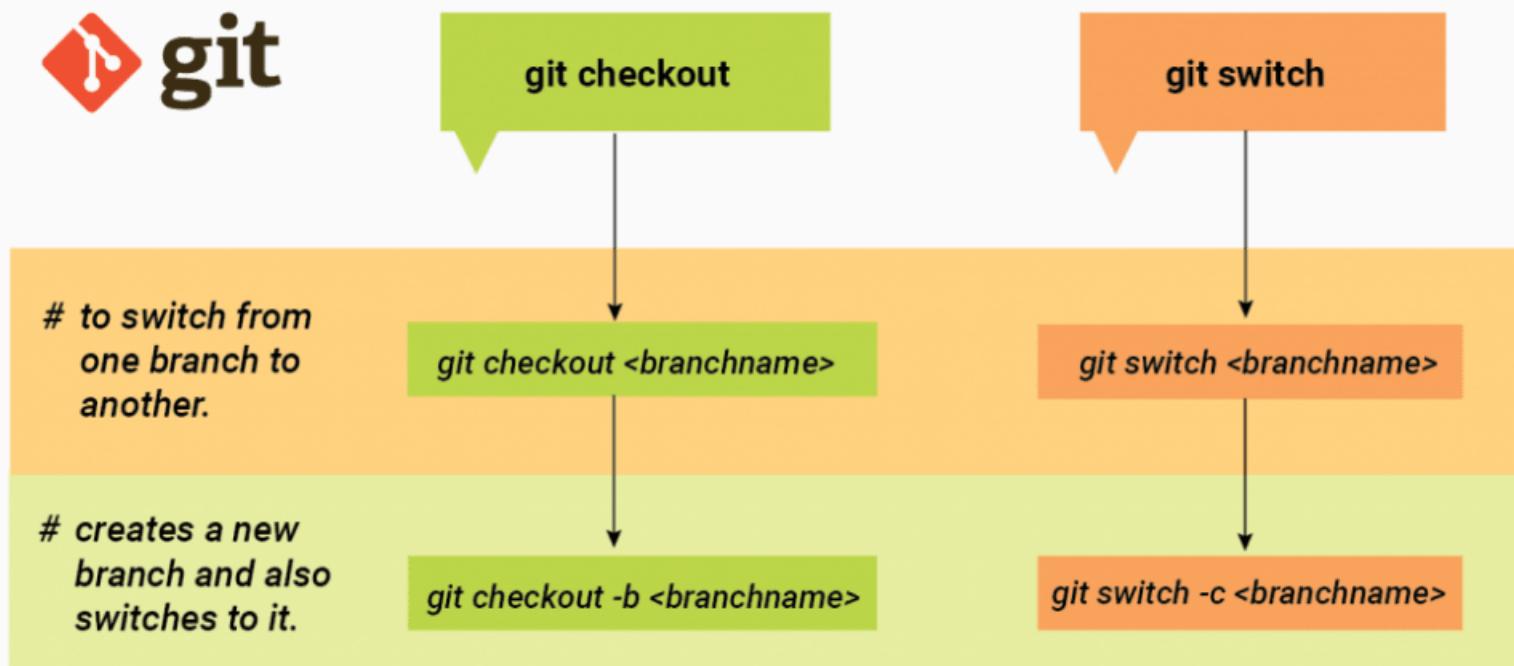
```
## branch-1  
## * master
```



1.10 Ramificações: branch, switch e merge

switch: troca a raficação de edição

```
git switch branch-1
```

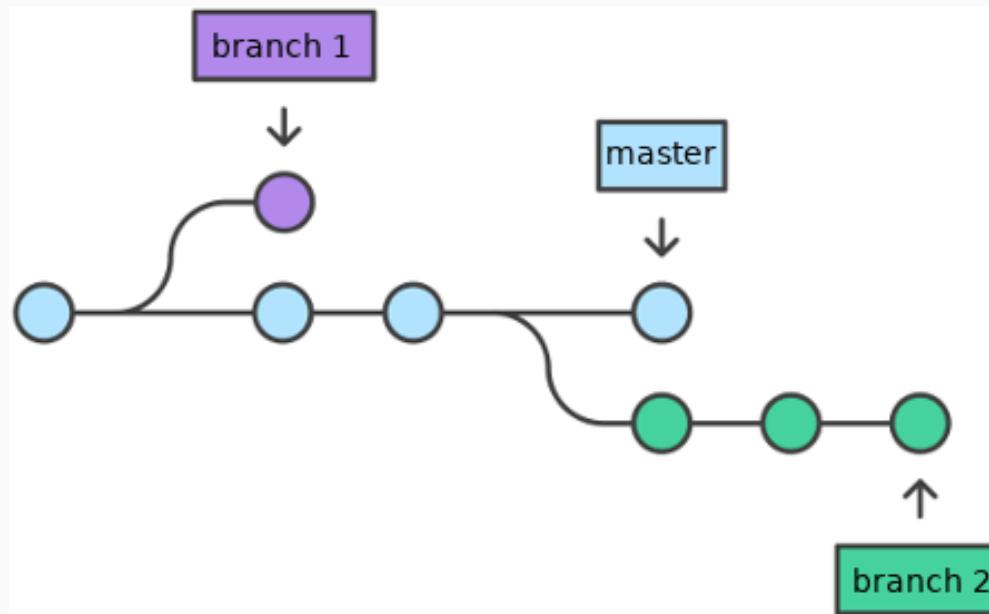


1.10 Ramificações: branch, switch e merge

branch: listar os branches e verificar o branch de edição

```
git branch
```

```
## * branch-1  
##   master
```



Vamos criar um arquivo no diretório

Usando a aba **Terminal** do RStudio

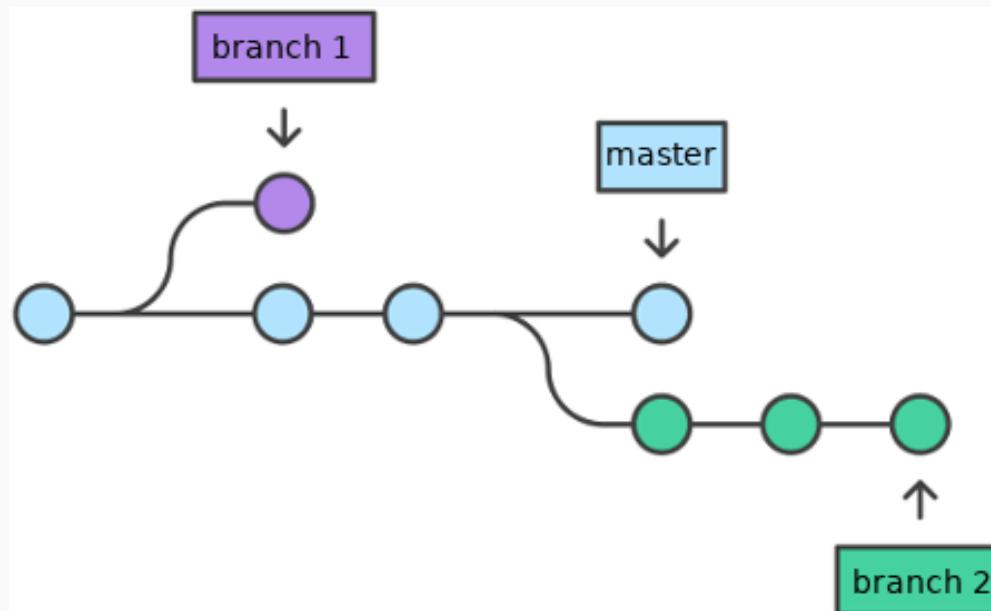
```
# terminal  
touch teste_branch1.txt  
git add -Av  
git commit -m "2020-10-19"
```



1.10 Ramificações: branch, switch e merge

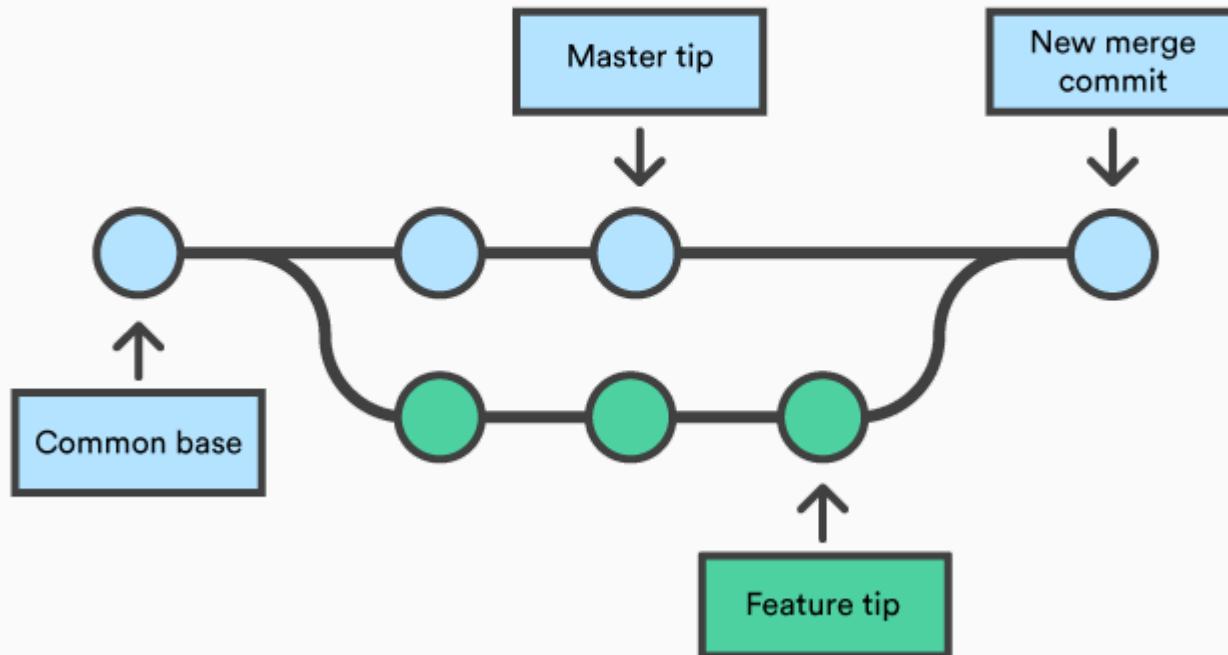
switch: troca a raficação de edição

```
git switch master
```



1.10 Ramificações: branch, switch e merge

merge: mescla as linhas de desenvolvimento independentes em um único branch



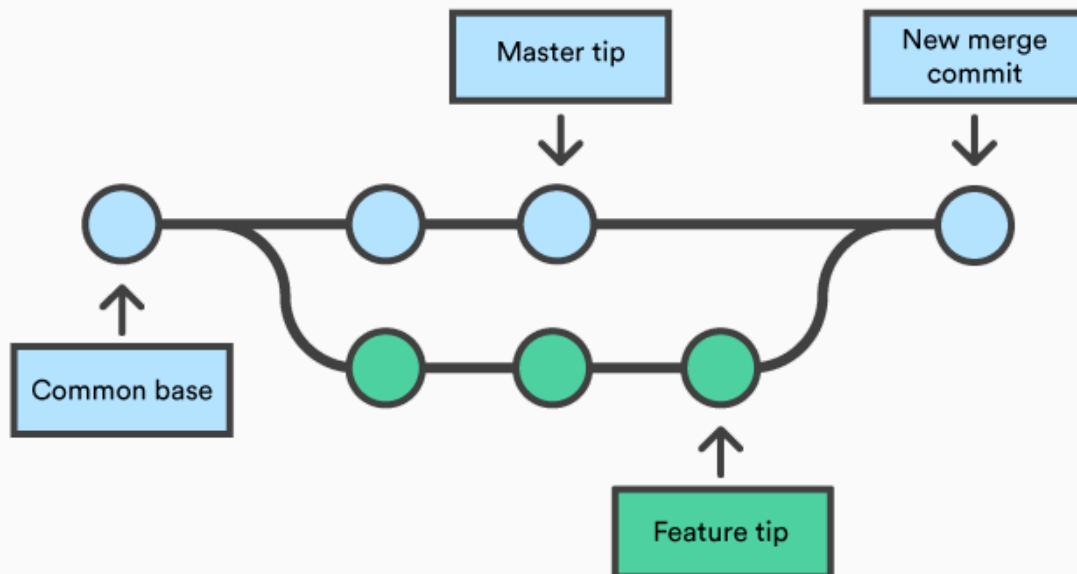
1.10 Ramificações: branch, switch e merge

merge: mescla as linhas de desenvolvimento independentes em um único branch

```
git merge branch-1
```

```
## teste_branch1.txt
```

```
| 0
```



1.10 Ramificações: branch, switch e merge

CUIDADO!



1.10 Ramificações: branch, switch e merge

CUIDADO!



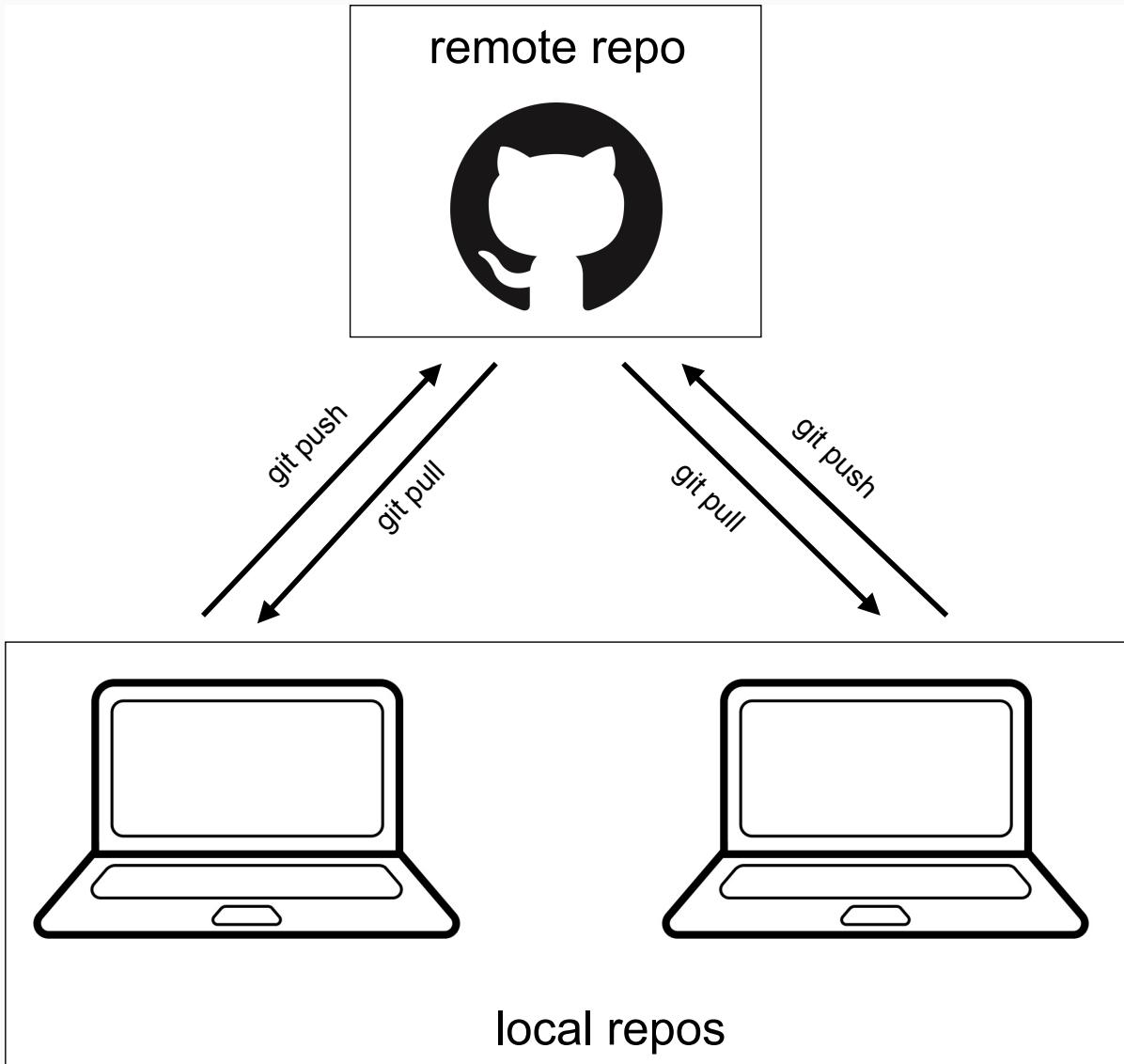
Origin

Master

>git merge

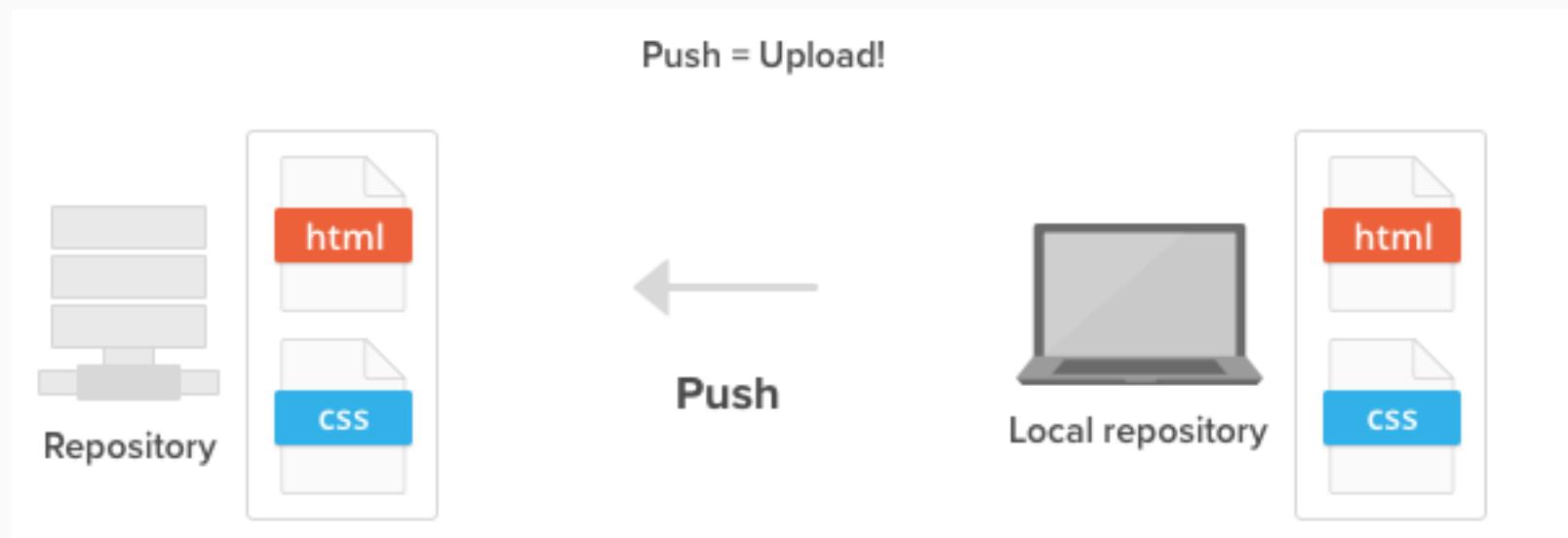


Merging would be like...



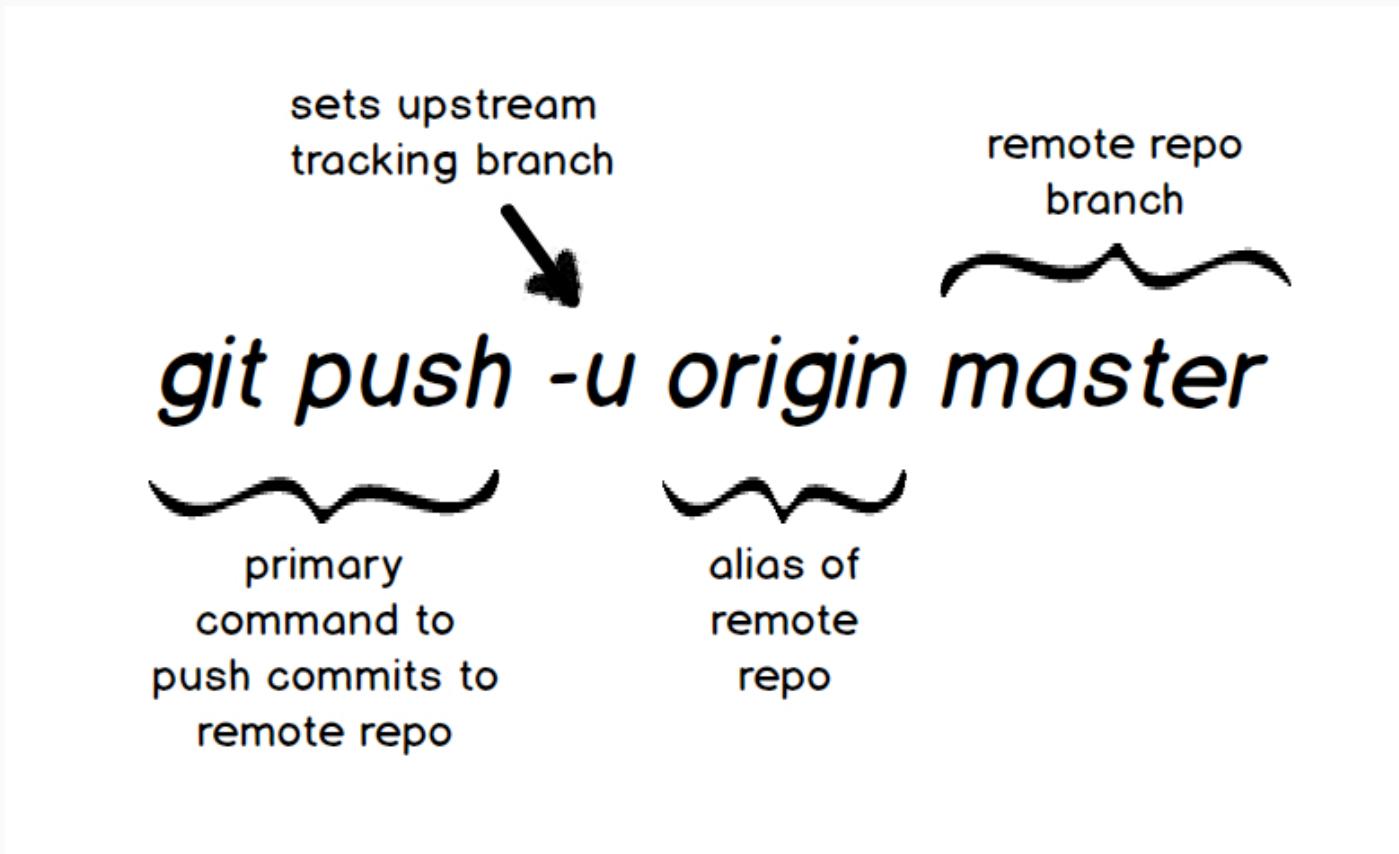
1.11 Remoto: push e pull

push: empurrar uma nova versão do repositório remoto (GitHub) para o repositório local



1.11 Remoto: push e pull

push: empurrar uma nova versão do repositório remoto (GitHub) para o repositório local



1.11 Remoto: push e pull

push: empurrar uma nova versão do repositório remoto (GitHub) para o repositório local

```
git push -u origin master
```

```
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 230 bytes | 230.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/mauriciovancine/disciplina-analise-geoespacial-r.git
  938e7b2..6ba8f7f  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

1.11 Remoto: push e pull

CUIDADO!



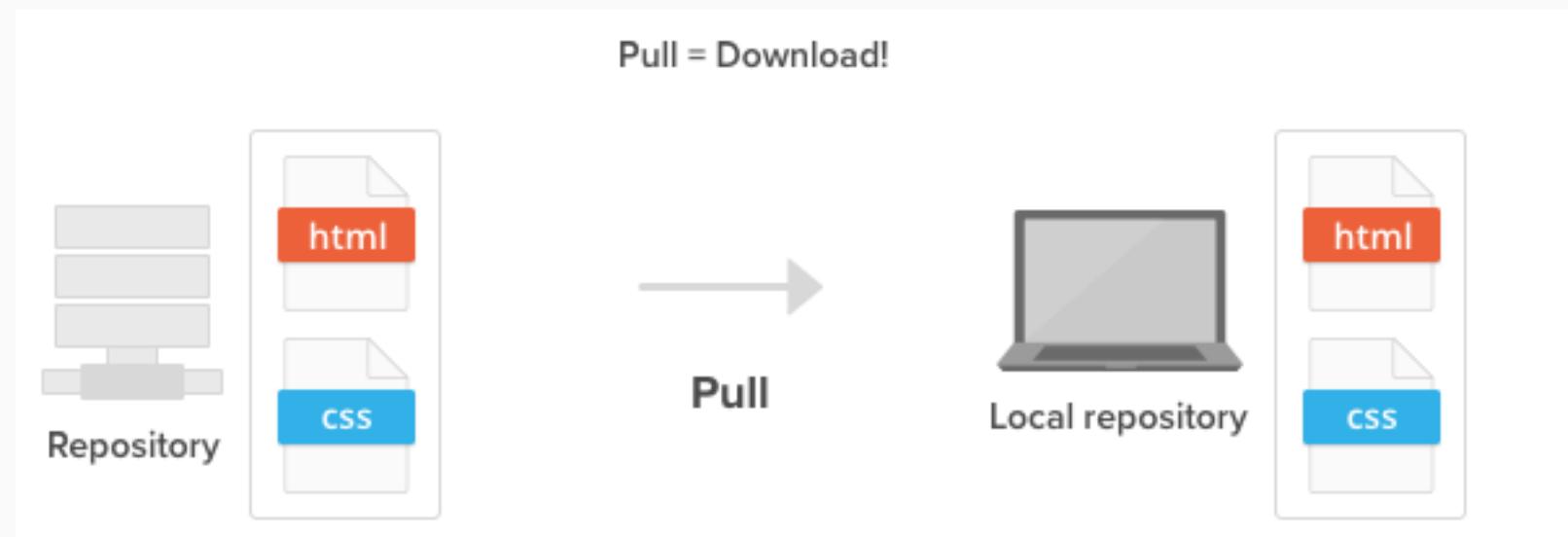
1.11 Remoto: push e pull

CUIDADO!



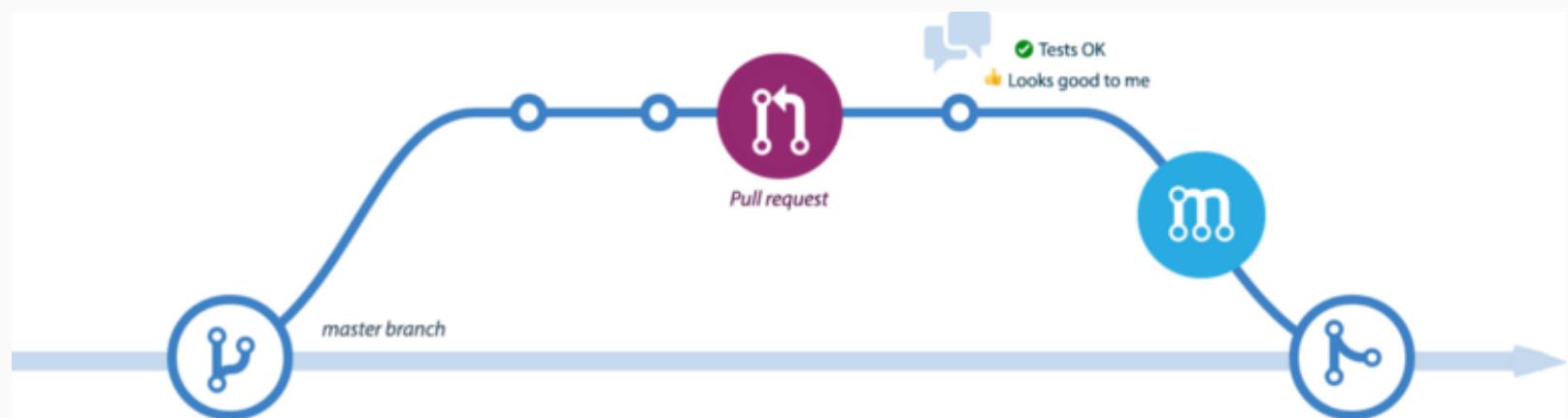
1.11 Remoto: push e pull

pull: puxar uma nova versão do repositório remoto (GitHub) para o repositório local



1.12 Pull request

Pull request: solicita que o repositório de destino (remoto) aceite ('puxe') suas alterações



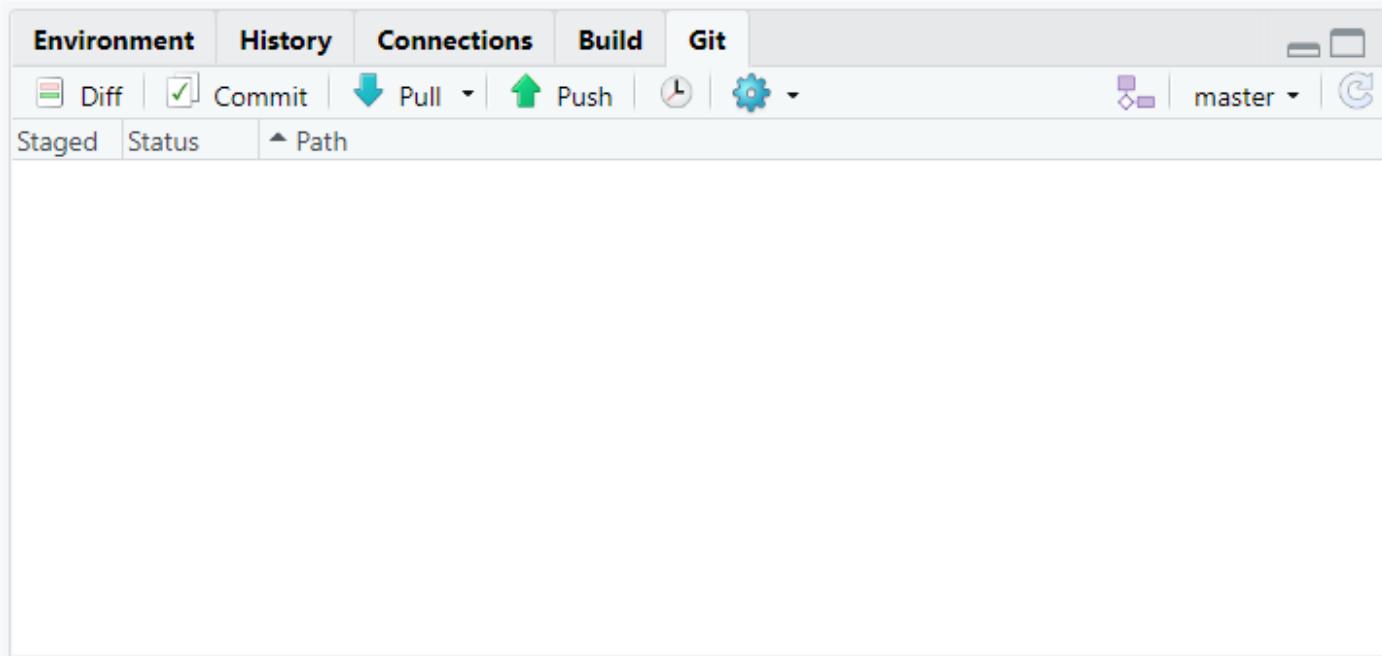
Interface Gráfica RStudio

1.13 Interface Gráfica RStudio

Git Panel

RStudio tem um **cliente Git** na aba "Git"

Esse painel aparece em **projetos que estejam versionados com git**

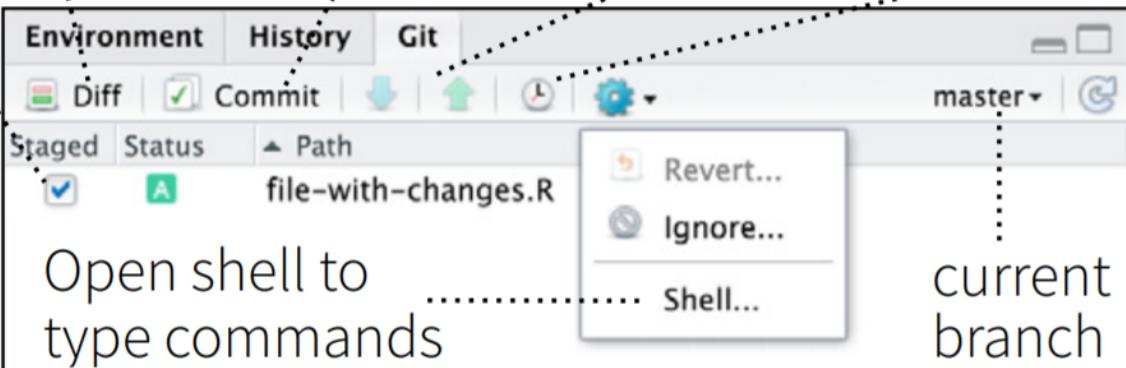


1.13 Interface Gráfica RStudio

Git Panel - Detalhes

Version Control with Git or SVN

Turn on at **Tools > Project Options > Git/SVN**



The screenshot shows the RStudio interface with the 'Git' tab selected in the top navigation bar. Below it, the 'Status' tab is active. A context menu is open over a file named 'file-with-changes.R', showing options like 'Revert...', 'Ignore...', and 'Shell...'. The 'Staged' tab is also visible. On the left, there's a legend for file status icons: green '+' for Added, red '-' for Deleted, blue 'M' for Modified, purple 'R' for Renamed, and yellow '?' for Untracked.

Icon	File Status
A	Added
D	Deleted
M	Modified
R	Renamed
?	Untracked

Stage files:

Show file diff

Commit staged files

Push/Pull to remote

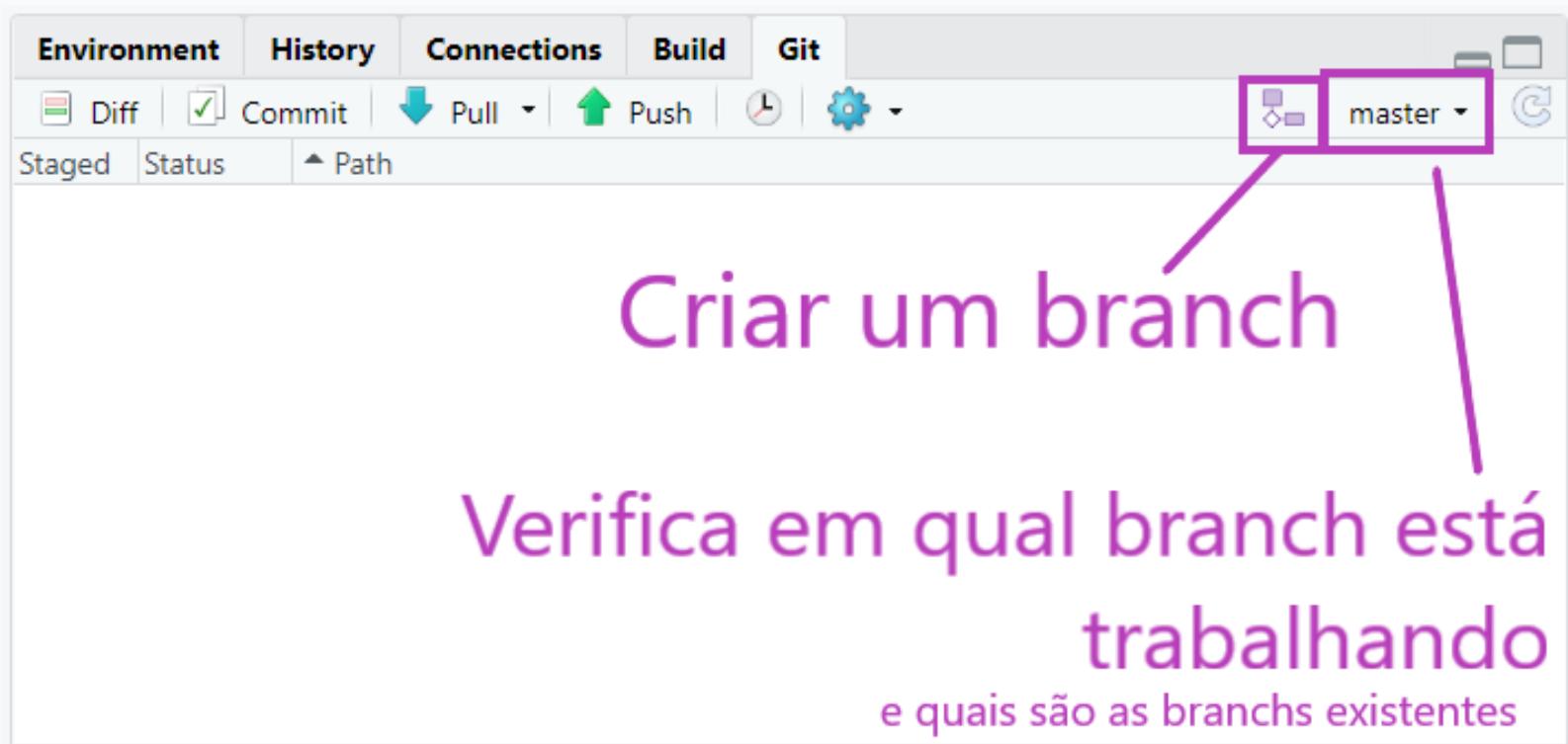
View History

current branch

Open shell to type commands

1.13 Interface Gráfica RStudio

Git Panel - Branches



1.13 Interface Gráfica RStudio

Git Panel - Diff - Changes: Revisar mudanças

The screenshot shows the RStudio interface with the 'Review Changes' panel open. The 'Changes' tab is selected, showing a list of files in the 'Staged' and 'Status' columns. The 'master' branch is selected. The 'Commit message' field is empty. Below the file list, there's a diff viewer with a context menu and buttons for 'Stage chunk' and 'Discard chunk'. The main pane displays the code changes:

```
@@ -532,36 +532,27 @@ knitr::include_graphics("img/gifs/create-proj2.gif")
532 532 ---
533 533
534 534 # Prática 3 - Fork + Clone
535 535
536 536 `~{r echo=TRUE, eval=FALSE}
537 537 usethis::create_from_github("rstudio-education/remaster-the-tidyverse", #usuário/repositório
538 538 usethis::create_from_github("rstudio-education/datascience-box", #usuário/repositório
539 539 destdir = "C:/Users/beatr/Documents/GitHub", #diretório onde quer que os arquivos sejam salvos
      fork = TRUE)
540 540 # creating 'C:/Users/beatr/Documents/GitHub/remaster-the-tidyverse/'
541 541 # ✓ Forking 'rstudio-education/remaster-the-tidyverse'
542 542 # Which git protocol to use? (enter 0 to exit)
543 543 #
544 544 # 1: ssh  <-- presumes that you have set up ssh keys
545 545 {{# 2: https <-- choose this if you don't have ssh keys (or don't know if you do)}}
546 546 #
547 547 # Selection: 2
548 548 # • Tip: To suppress this menu in future, put
549 549 #   `options(usethis.protocol = "https")`
550 550 #   in your script or in a user- or project-level startup file, '.Rprofile'.
551 551 #   call `usethis::edit_r_profile()` to open it for editing.
```

1.13 Interface Gráfica RStudio

Git Panel - Diff - History: Histórico de mudanças

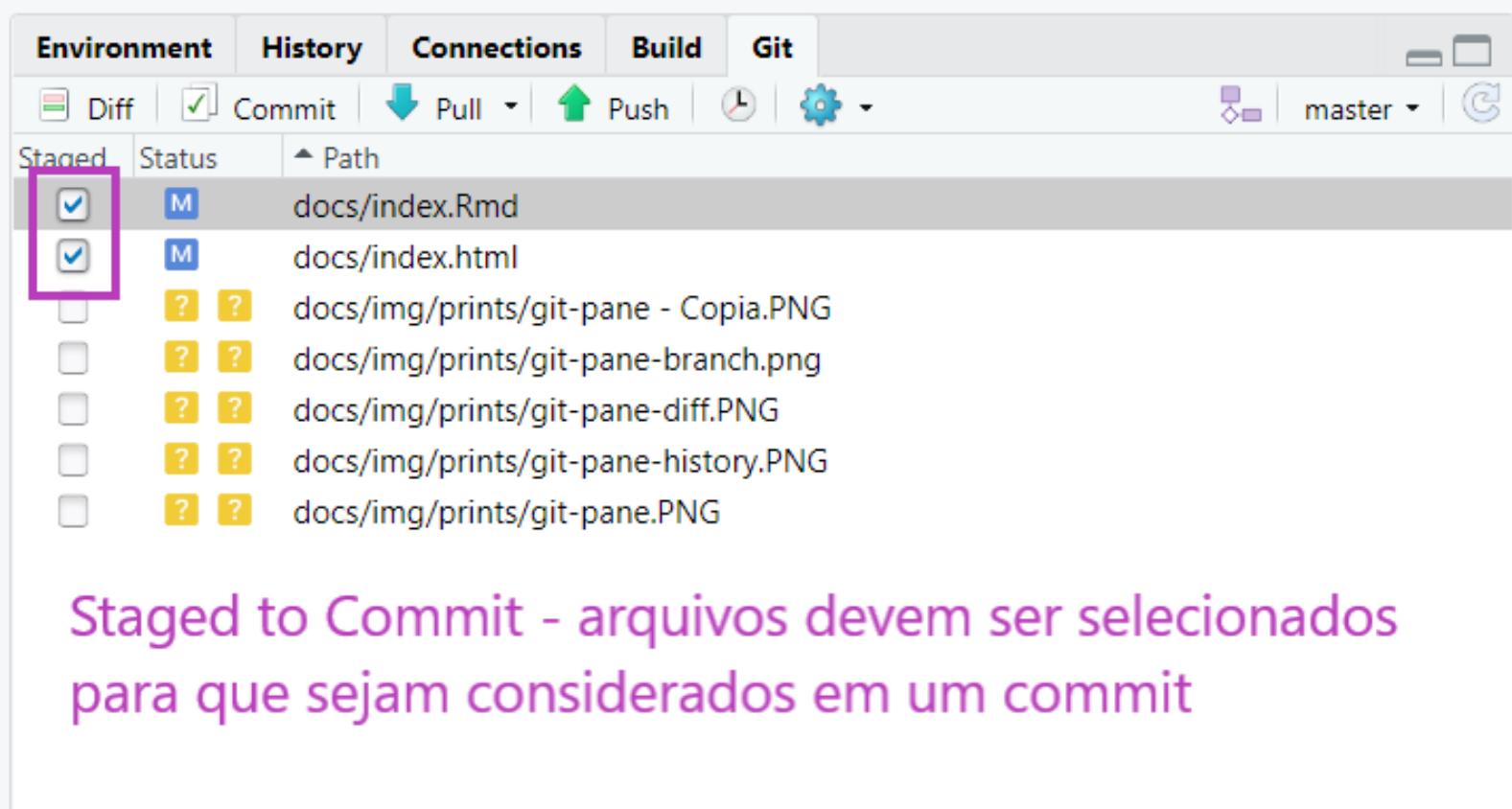
The screenshot shows the RStudio interface with the following components visible:

- Git Panel (Top Bar):** Shows "Changes" selected, "History" tab, "master" branch, and "(all commits)" dropdown.
- History Tab:** Displays a list of commits from "origin/master".

Author	Date	SHA
Beatriz Milz <beatriz.milz@hotmail.com>	2019-08-29	ab5bd162
Beatriz Milz <beatriz.milz@hotmail.com>	2019-08-29	0449805c
Beatriz Milz <beatriz.milz@hotmail.com>	2019-08-29	85a5479e
Beatriz Milz <beatriz.milz@hotmail.com>	2019-08-29	c739f6ab
Beatriz Milz <beatriz.milz@hotmail.com>	2019-08-29	babc8bfe
Beatriz Milz <beatriz.milz@hotmail.com>	2019-08-29	06c6f018
Beatriz Milz <beatriz.milz@hotmail.com>	2019-08-29	06211470
- Diff View (Bottom):** Shows the diff for the file "docs/index.Rmd". The changes are color-coded:
 - Red (deletions): "@@ -96,10 +96,10 @@ class: split-33 with-border", "## Pré-requisitos", "97 97]]", "98 98 .row.bg-main2[.content[", "99 99 ## Configurando o `Git`", "100 100 ## Configurando o Git", "101 101]]", "102 102 .row.bg-main3[.content[", "103 103 ## Configurando o `GitHub`", "104 104 ## Configurando o GitHub", "105 105 .row.bg-main4[.content[", "106 106 ## Trabalhando com projetos no `Rstudio` + `GitHub`", "107 107 @@ -129,7 +129,7 @@ class: middle", "108 108 # Pré-requisitos", "109 109]]".
 - Green (additions): "99 99 ## Configurando o Git", "100 100 ## Configurando o Git", "101 101 .row.bg-main3[.content[", "102 102 .row.bg-main3[.content[", "103 103 ## Configurando o `GitHub`", "104 104 ## Configurando o GitHub", "105 105 .row.bg-main4[.content[", "106 106 ## Trabalhando com projetos no `Rstudio` + `GitHub`", "107 107 @@ -129,7 +129,7 @@ class: middle", "108 108 # Pré-requisitos", "109 109]]".

1.13 Interface Gráfica RStudio

Git pane - Staged



The screenshot shows the RStudio interface with the 'Git' tab selected in the top navigation bar. Below the navigation bar is a toolbar with icons for Diff, Commit, Pull, Push, and settings. To the right of the toolbar is a dropdown showing 'master'. The main area is the 'Git pane' titled 'Staged'. It lists several files with their status indicators:

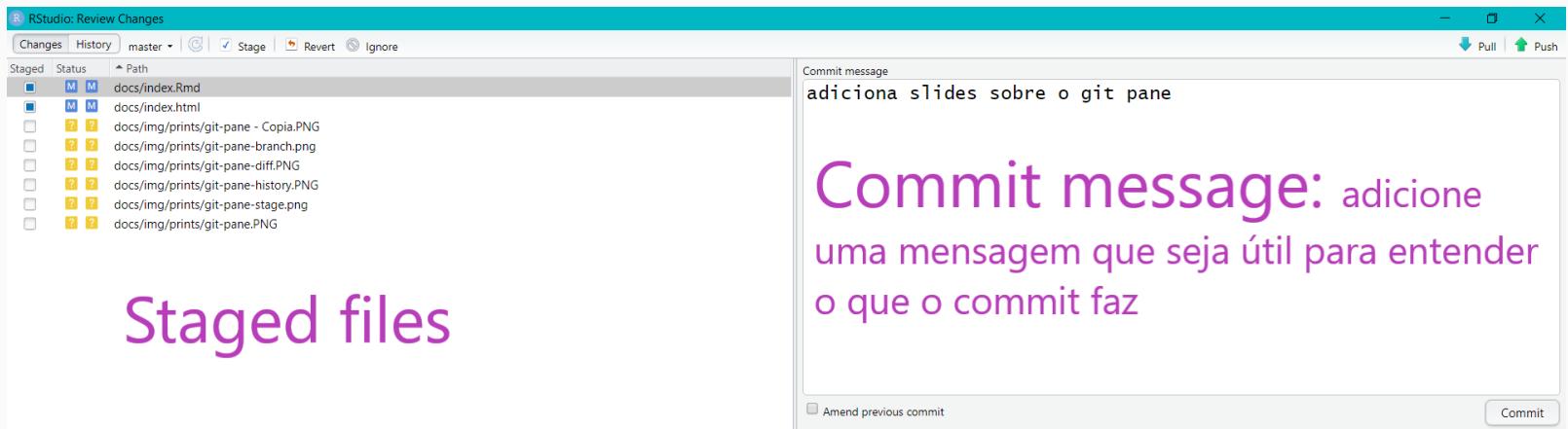
			Path
<input checked="" type="checkbox"/>	M		docs/index.Rmd
<input checked="" type="checkbox"/>	M		docs/index.html
<input type="checkbox"/>	?	?	docs/img/prints/git-pane - Copia.PNG
<input type="checkbox"/>	?	?	docs/img/prints/git-pane-branch.png
<input type="checkbox"/>	?	?	docs/img/prints/git-pane-diff.PNG
<input type="checkbox"/>	?	?	docs/img/prints/git-pane-history.PNG
<input type="checkbox"/>	?	?	docs/img/prints/git-pane.PNG

A purple box highlights the first two rows where the 'Staged' checkbox is checked.

Staged to Commit - arquivos devem ser selecionados para que sejam considerados em um commit

1.13 Interface Gráfica RStudio

Git pane - Commit



Staged files

The screenshot shows the RStudio interface with the 'Review Changes' tab selected. In the top right, there are 'Pull' and 'Push' buttons. Below them, a 'Commit message' field contains the text 'adiciona slides sobre o git pane'. On the left, a tree view shows several files under the 'docs' directory, with some being tracked (M) and others unstaged (D). A large pink box on the right contains the text: 'Commit message: adicione uma mensagem que seja útil para entender o que o commit faz'.

Show Staged Unstaged Context 5 line Ignore Whitespace Stage All Discard All

```
@@ -610,10 +610,20 @@ class: middle
610 610 # Botão `r emoji("clock3")` (history) -> Review changes
611 611 ```{r, out.width="90%"}
612 612 knitr::include_graphics("img/prints/git-pane-history.PNG")
613 613 ```

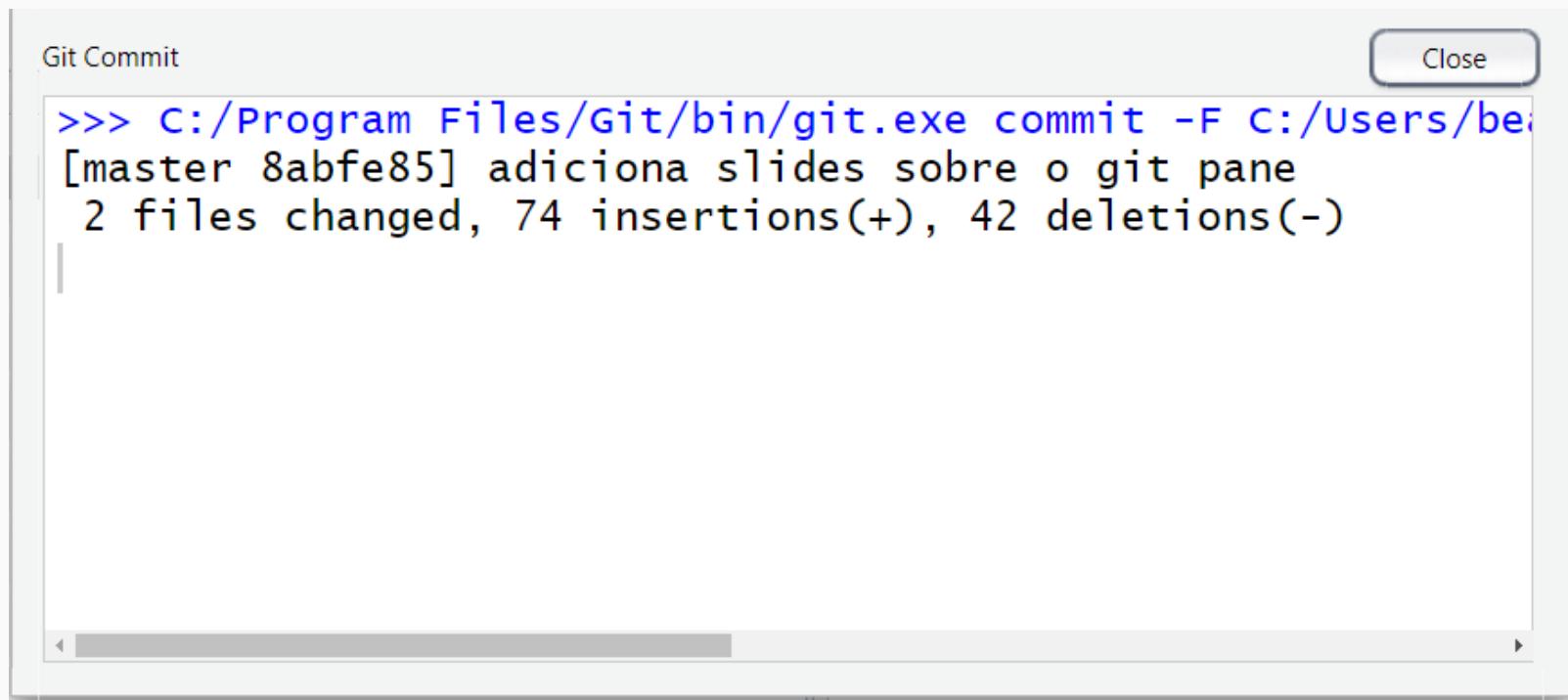
614 614
615
616
617 ---
618 class: middle
619
620 # Git pane - staged `r emoji("white_check_mark")`
621
622 ```{r}
623 knitr::include_graphics("img/prints/git-pane-stage.png")
624 ````
```

Stage chunk Discard chunk

Mudanças verificadas: em verde, são novas linhas. em vermelho, são linhas retiradas/alteradas

1.13 Interface Gráfica RStudio

Git pane - Commit



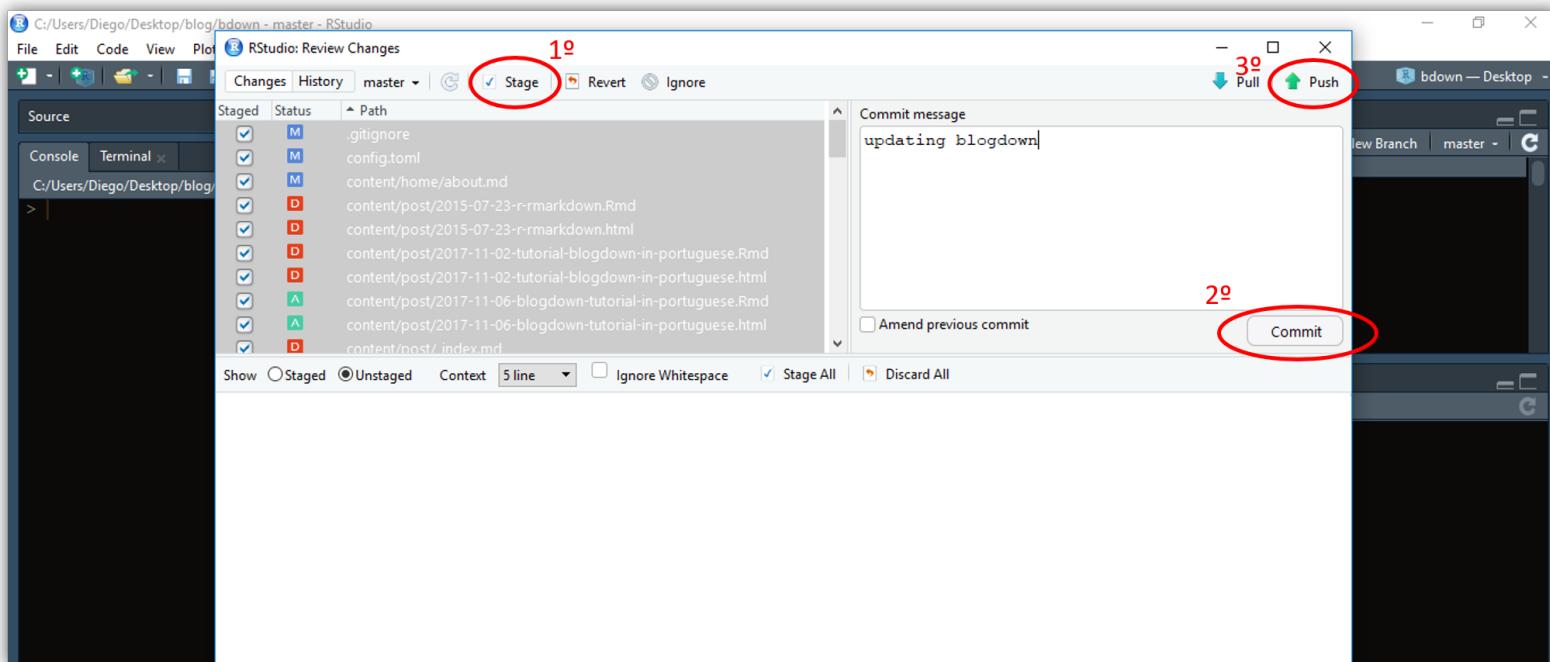
A screenshot of a 'Git Commit' dialog box from RStudio. The title bar says 'Git Commit' and there is a 'Close' button in the top right corner. The main area contains a command-line interface output:

```
>>> C:/Program Files/Git/bin/git.exe commit -F C:/Users/bebe/OneDrive/Área de Trabalho/slides.Rmd  
[master 8abfe85] adiciona slides sobre o git pane  
 2 files changed, 74 insertions(+), 42 deletions(-)
```

The output shows the command run, the commit message, and the statistics for the commit.

1.13 Interface Gráfica RStudio

Git pane - Push e Pull



Mais informações

Git Cheat Sheets



Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. This cheat sheet summarizes commonly used Git command line instructions for quick reference.

INSTALL GIT	MAKE CHANGES
GitHub provides desktop clients that include a graphical user interface for the most common repository actions and an automatically updating commandline edition of Git for advanced scenarios.	\$ git status Review edits and craft a commit transaction
GitHub for Windows https://windows.github.com	\$ git diff Lists all new or modified files to be committed
GitHub for Mac https://mac.github.com	\$ git add [file] Shows file differences not yet staged
Git distributions for Linux and POSIX systems are available on the official Git SCM web site.	\$ git diff --staged Schedules the file in preparation for versioning
Git for All Platforms http://git-scm.com	\$ git reset [file] Shows file differences between staging and the last file version
CONFIGURE TOOLING Configure user information for all local repositories	\$ git reset --hard Unstages the file, but preserve its contents
\$ git config --global user.name "[name]" Sets the name you want attached to your commit transactions	\$ git commit -m "[descriptive message]" Records file snapshots permanently in version history
\$ git config --global user.email "[email address]" Sets the email you want attached to your commit transactions	
\$ git config --global color.ui auto Enables helpful colorization of command line output	
CREATE REPOSITORIES	GROUP CHANGES
Start a new repository or obtain one from an existing URL	Name a series of commits and combine completed efforts
\$ git init [project-name] Creates a new local repository with the specified name	\$ git branch Lists all local branches in the current repository
\$ git clone [url] Downloads a project and its entire version history	\$ git branch [branch-name] Creates a new branch
	\$ git checkout [branch-name] Switches to the specified branch and updates the working directory
	\$ git merge [branch] Combines the specified branch's history into the current branch
	\$ git branch -d [branch-name] Deletes the specified branch

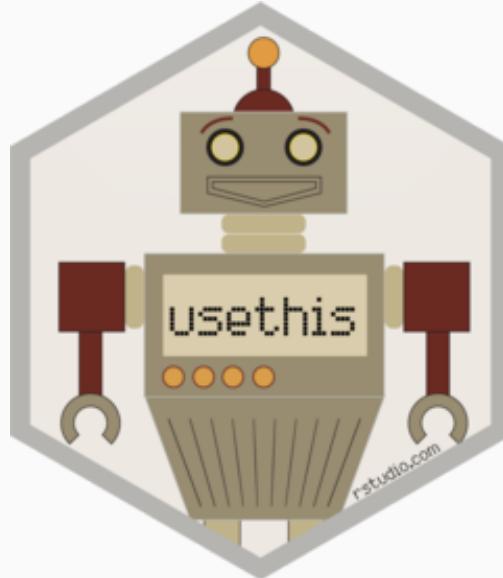
[*] https://github.github.com/training-kit/downloads/pt_BR/github-git-cheat-sheet.pdf

Mais informações

Pacote usethis

Comandos direto no R para criar e versionar um repositório

```
install.packages("usethis")
devtools::install_github("r-lib/usethis")
```



Mais informações

Material

[Pro Git](#) - Scott Chacon e Ben Straub

[Happy Git and GitHub for the useR](#) - Jenny Bryan

[Primeiros passos utilizando o Git e GitHub no RStudio](#) - Beatriz Milz

[Oh Shit, Git!?!](#) - Katie Sylor-Miller

[Dangit, Git!?!](#) - Katie Sylor-Miller

Dúvidas?

Maurício Vancine

Contatos:

 mauricio.vancine@gmail.com

 [@mauriciovancine](https://twitter.com/mauriciovancine)

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

 mauriciovancine.netlify.com

Slides criados via pacote [xaringan](#) e tema [Metropolis](#)