

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)

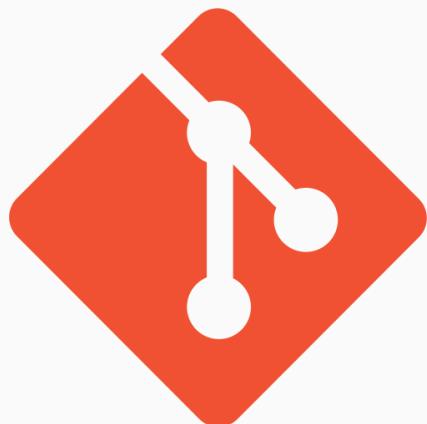
<https://www.rstudio.com/>



1.1 Verificar os notebooks

git (>2.28)

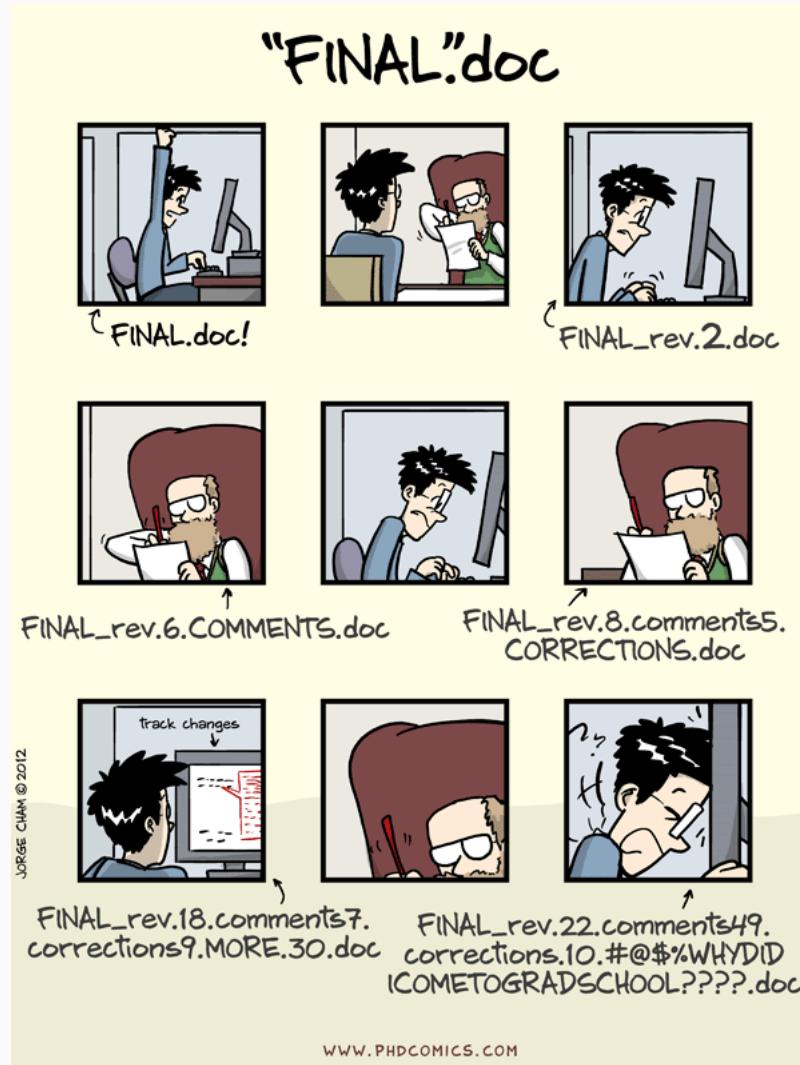
<https://git-scm.com/>



git

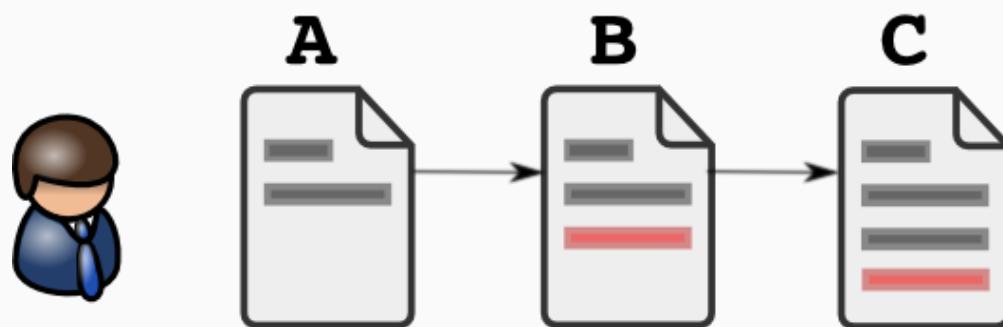
Tudo instalado? Então bora!

1.2 Controle de versão



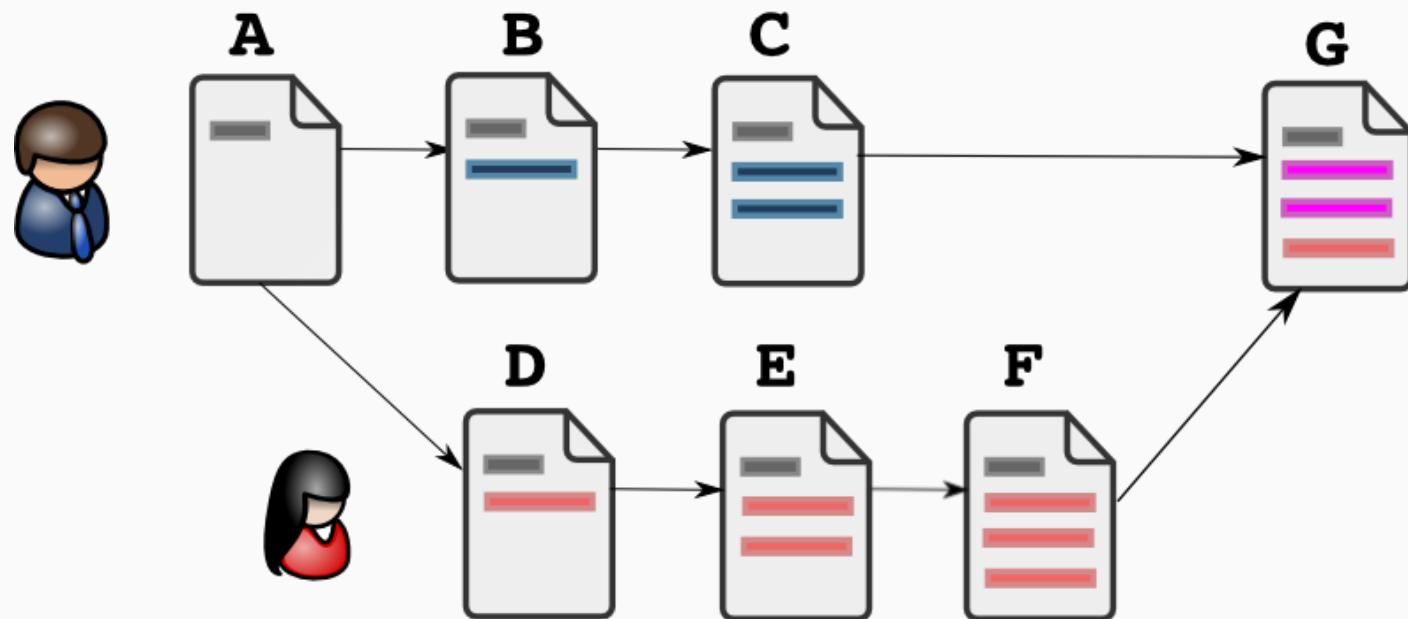
1.2 Controle de versão

Manejar projetos **individualmente**



1.2 Controle de versão

Manejar projetos **compartilhados**



1.3 Git

Software que faz o **controle de versão**

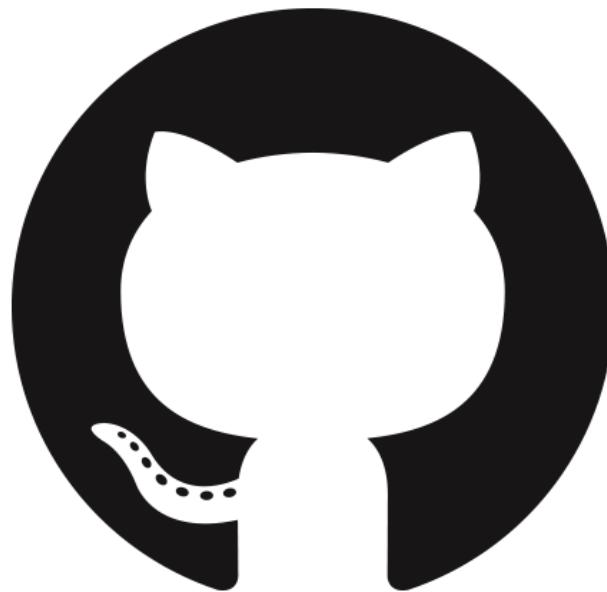
Maneja os repositórios locais (e.g. notebook) e remotos (e.g. GitHub)



1.4 GitHub

Repositório remoto

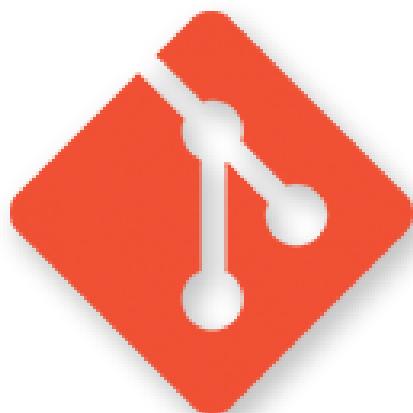
Plataforma de hospedagem de código-fonte com controle de versão usando o Git



ATENÇÃO!

Git: software que faz o controle de versão

GitHub: repositório remoto que hospeda os arquivos versionados



git

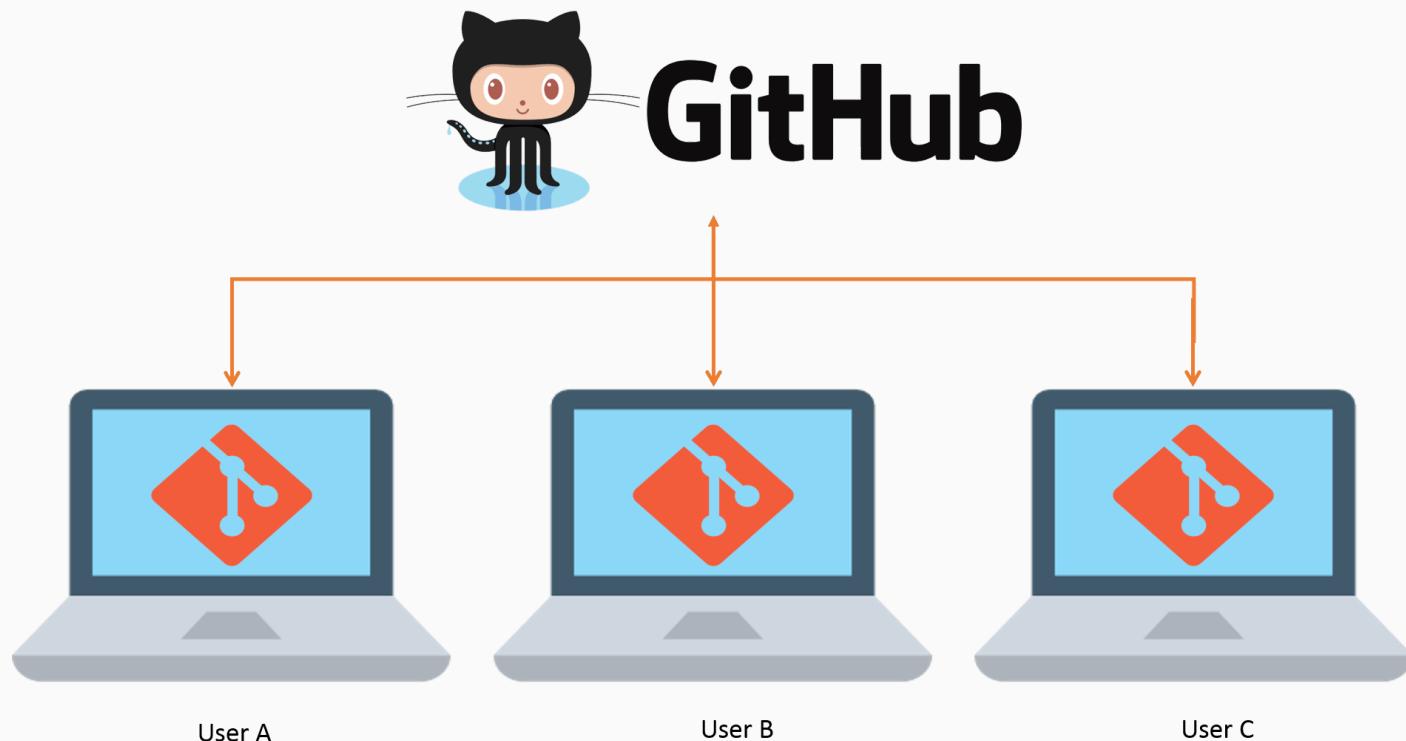


github
SOCIAL CODING

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!!!

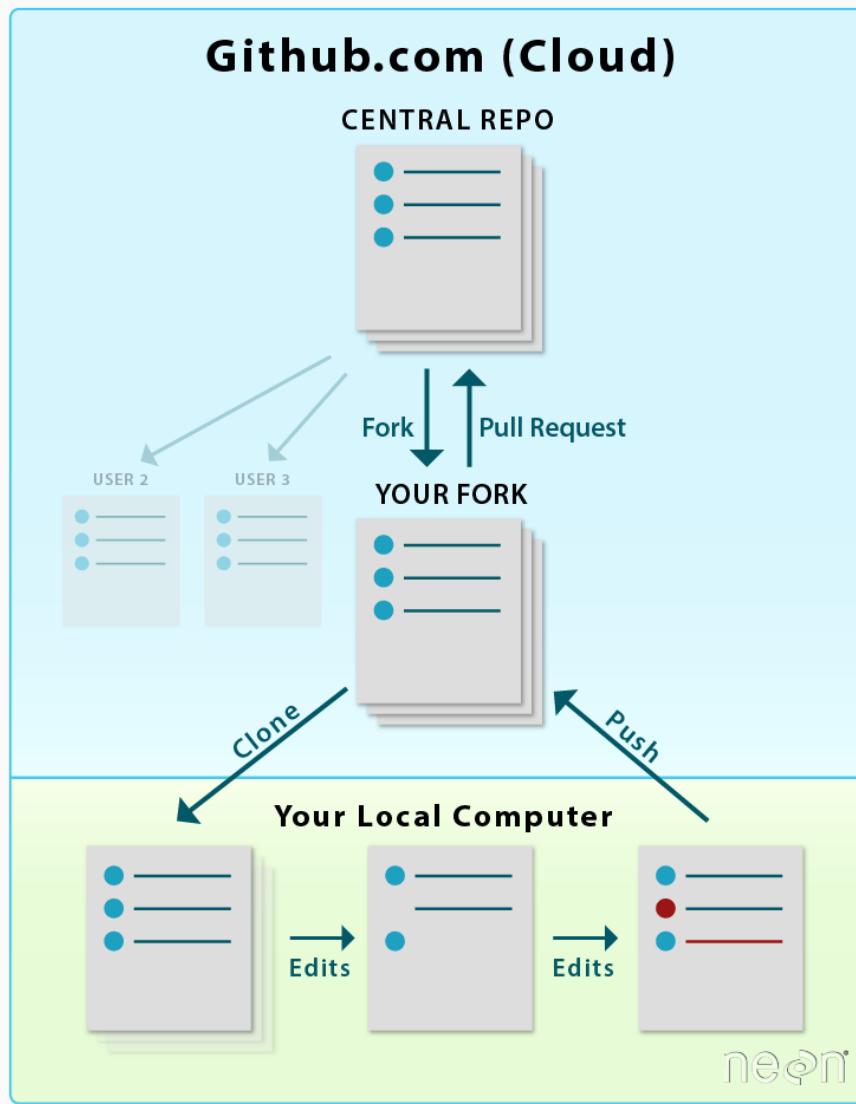
username: mauriciovancine

email: mauricio.vancine@gmail.com

senha: !@#\$%^&*+

Tudo bem até aqui?

1.5 Funcionamento do controle de



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

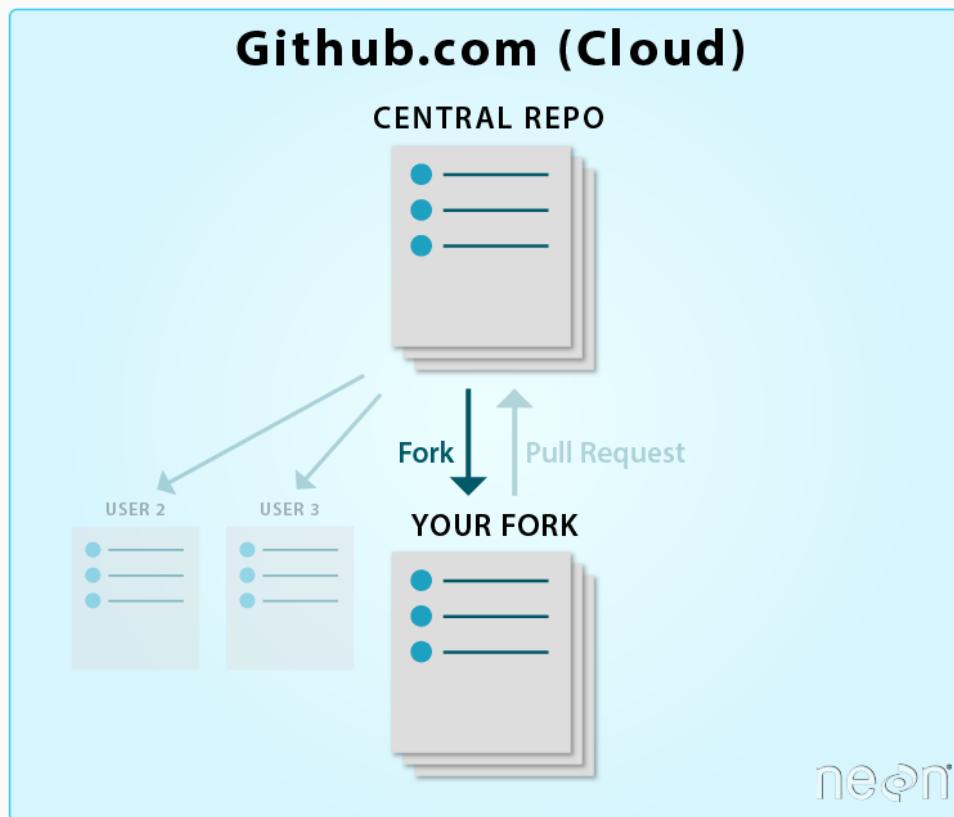
COOL. HOW DO WE USE IT?

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

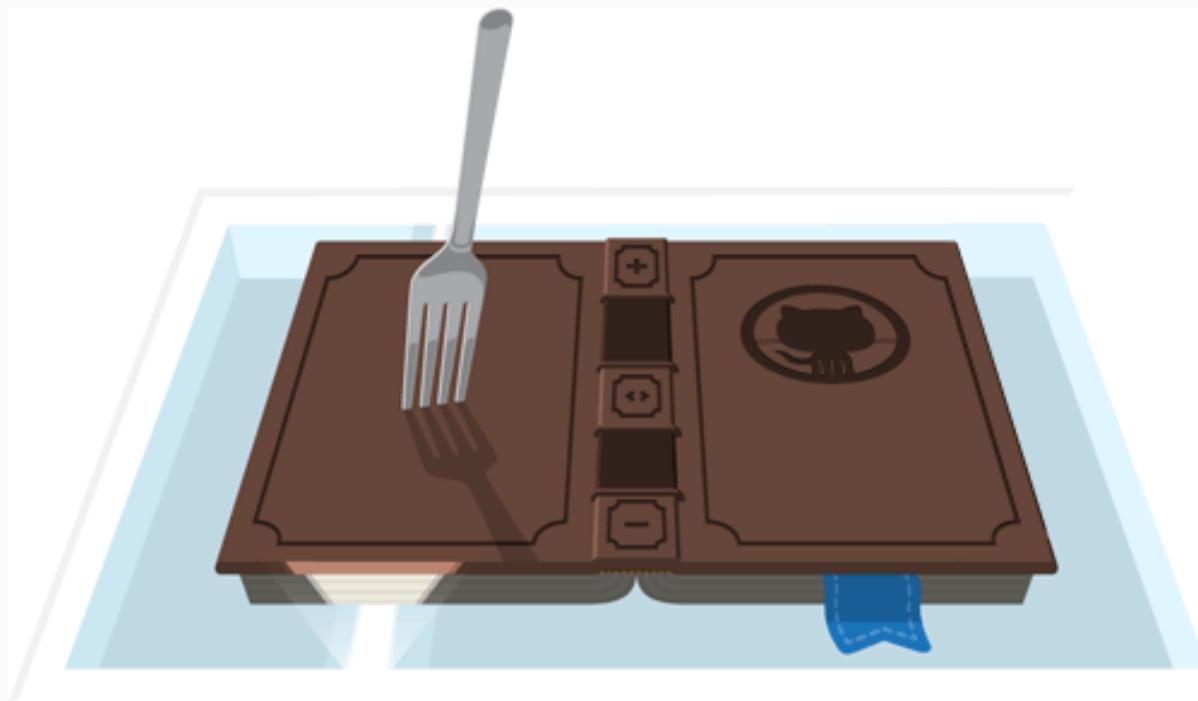
<https://github.com/mauriciovancine/disciplina-analise-geoespacial-r>

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. The 'Code' tab is selected. In the top right corner, there are buttons for Unwatch (with 1), Star (with 0), and Fork (with 1). The 'Fork' button is highlighted with a red box. Below the navigation, there are buttons for Go to file, Add file, and Code. The repository details show it was created by 'mauriciovancine' on 2020-10-07, with commit 83641ff made 11 minutes ago. The repository contains 1 branch and 0 tags. 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.' It also lists 'Readme', 'Releases' (no releases published), and 'Packages' (no packages published). The bottom right corner of the page has a gear icon.

1.6 Fork

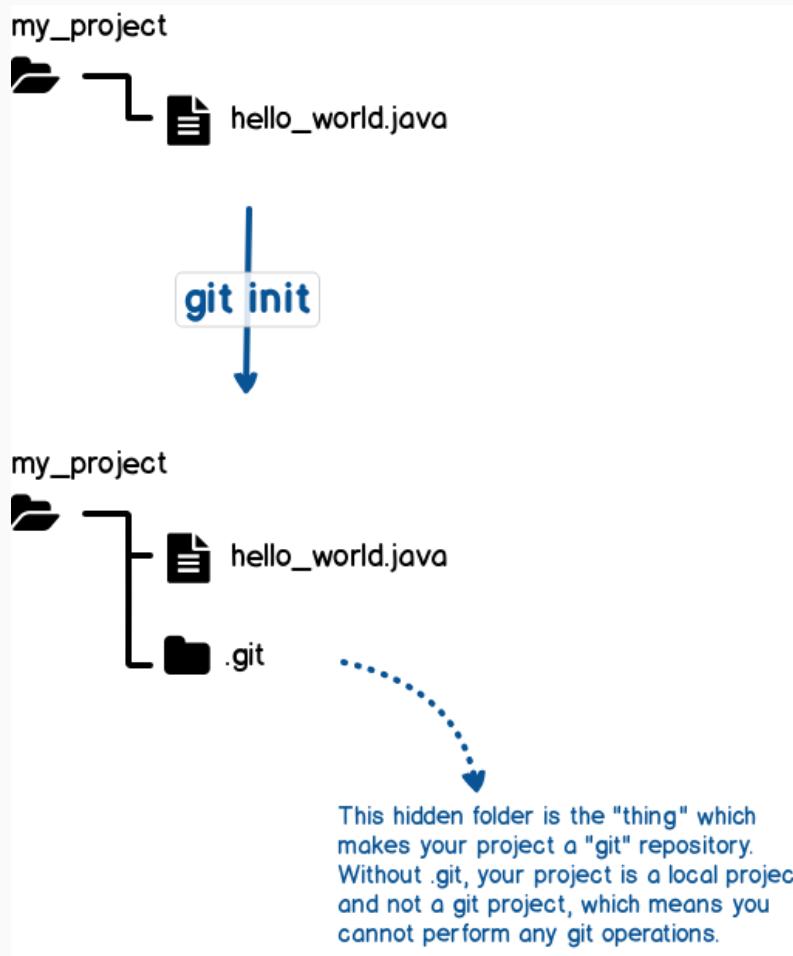
Forken o repositório da disciplina

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



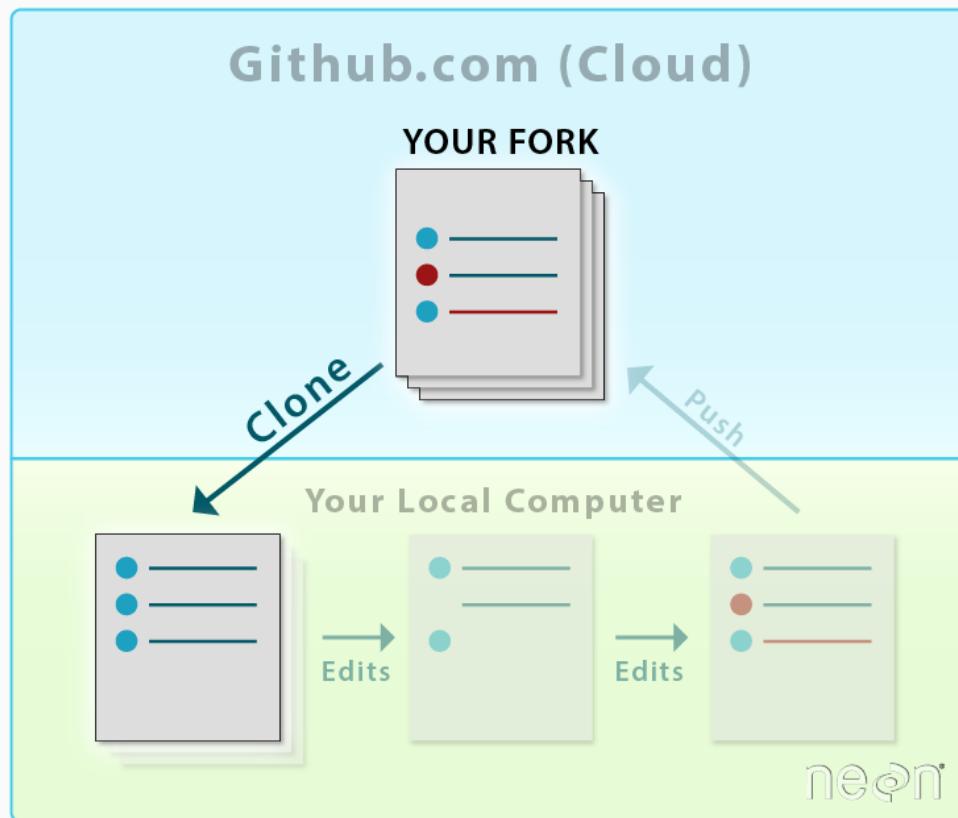
1.7 Iniciando: init ou clone

init: inicia o versionamento de um repositório



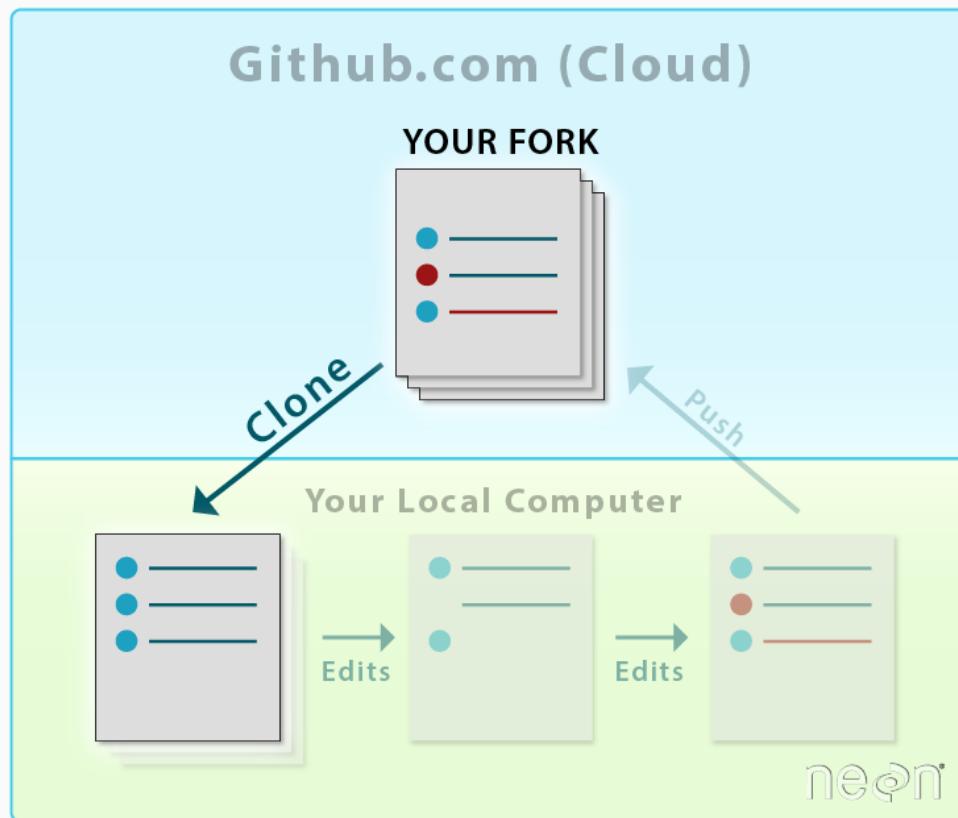
1.7 Iniciando: init ou clone

clone: copia um repositório remoto para o seu PC



1.7 Iniciando: init ou clone

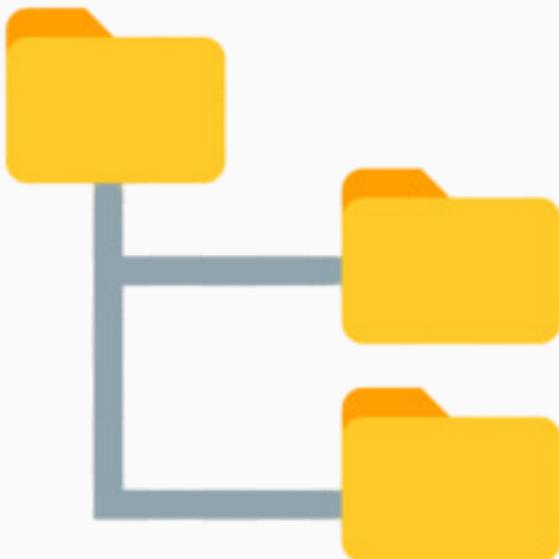
Vamos clonar o **repositório remoto (GitHub)** para o **repositório local (PC)**



1.7 Iniciando: init ou clone

Criem uma pasta chamada `github` no PC de vocês

```
".          # raiz"
"   ┌── home/      # home"
"   │   ┌── data/    # dados"
"   │   └── github/  # todos os repositorios"
```



E como faremos isso?

1.7 Iniciando: init ou clone

1. Download diretamente do repositório remoto no formato .zip

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. The repository details show 1 branch and 0 tags. The main content area displays a file tree with commits from 'mauriciovancine' on 2020-10-07. A red box highlights the 'Code' dropdown menu, which contains options for Clone (HTTPS, SSH, GitHub CLI) and a 'Download ZIP' button. Another red box highlights the 'Download ZIP' button. To the right, there are sections for About (no description), Releases (no releases published), and Packages (no packages published).

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

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: init ou clone

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, and the 'Clone' section is highlighted with a red box. The URL 'https://github.com/mauriciovancine/' is visible in the 'Clone' field.

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

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: init ou clone

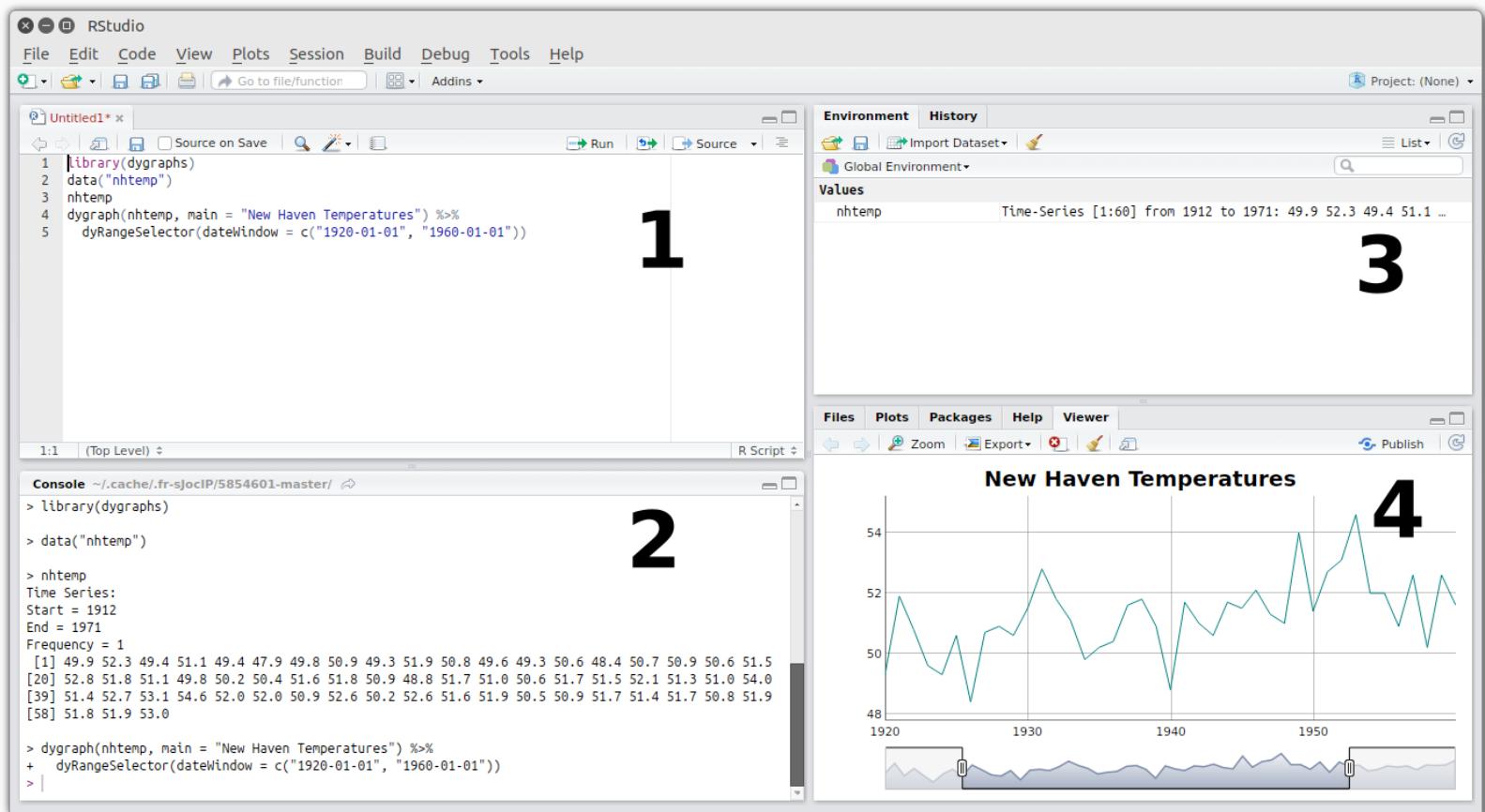
3. Usar o **RStudio** para clonar pelo **HTTPS** do repositório



Vamos abrir o RStudio

RStudio

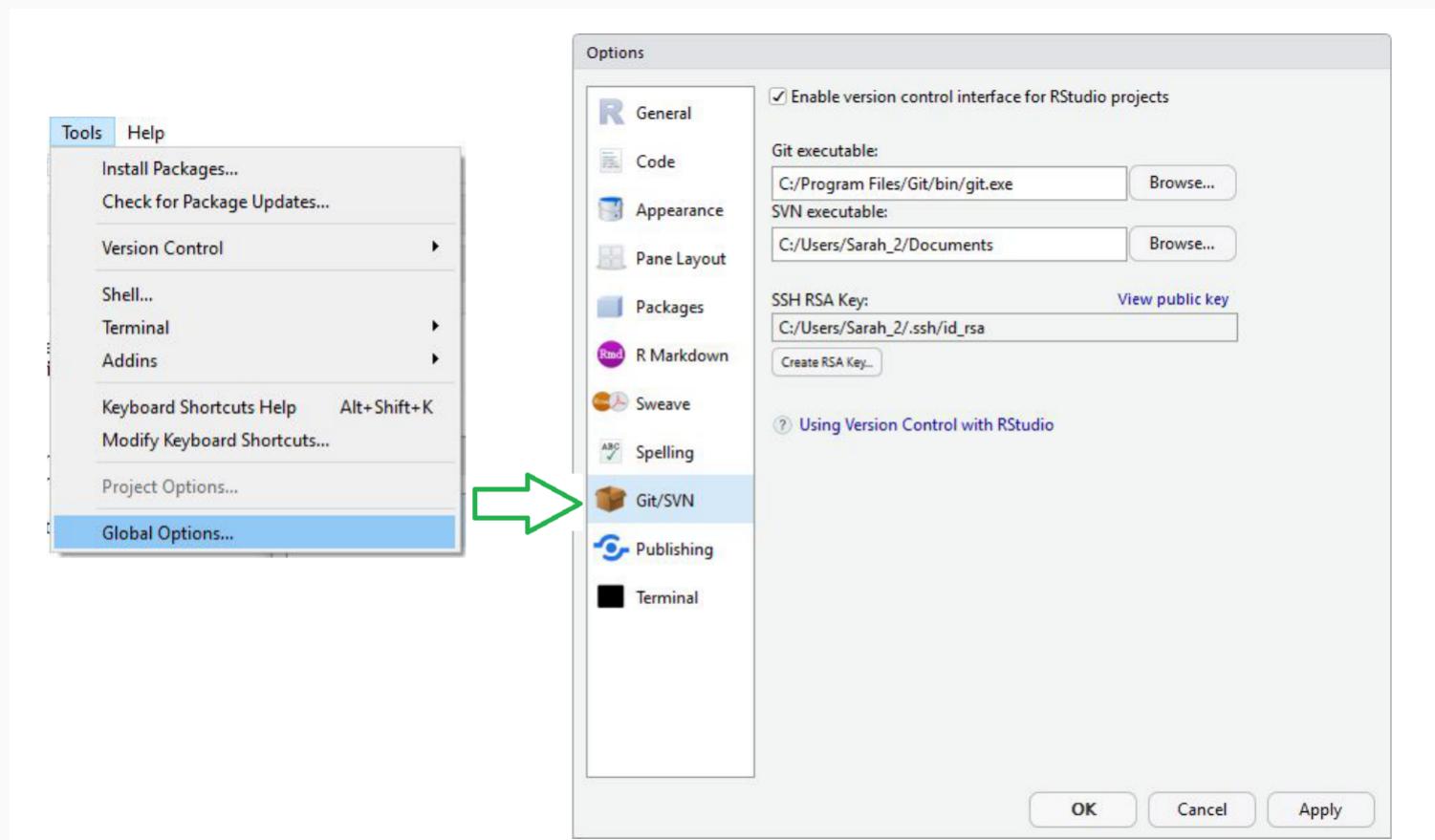
IDE RStudio



Git e RStudio

Indicar o caminho do git para o RStudio

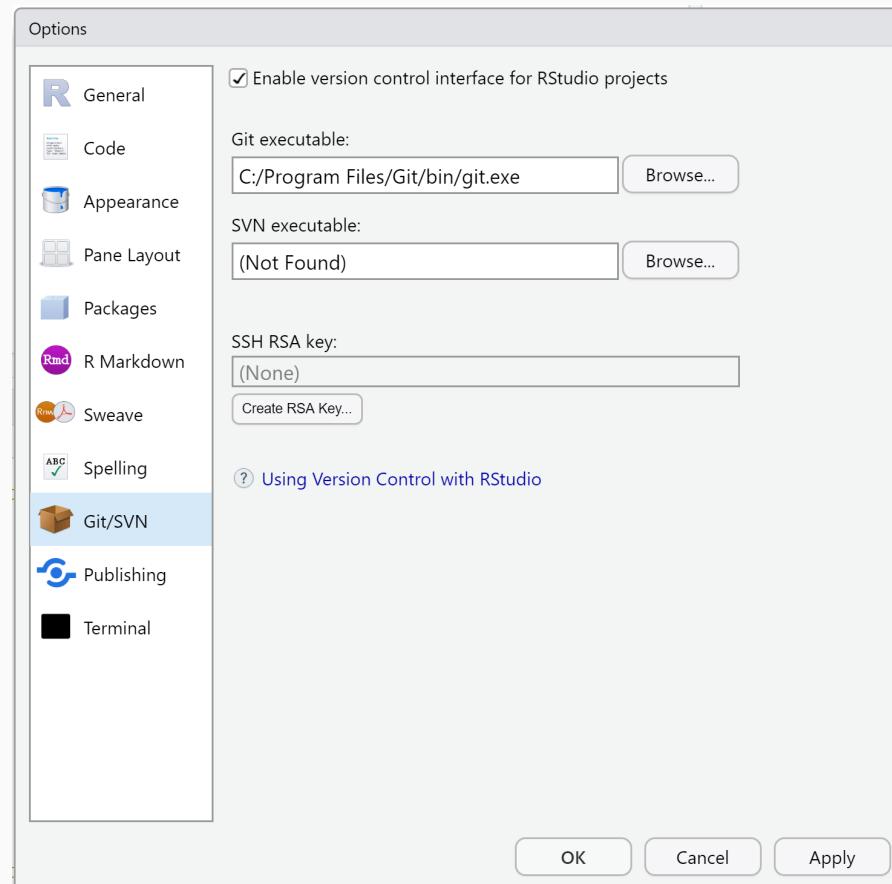
- Tools > Global Options > Git/SVN



Git e RStudio

Marcar **Enable version...** e **Git executable:**

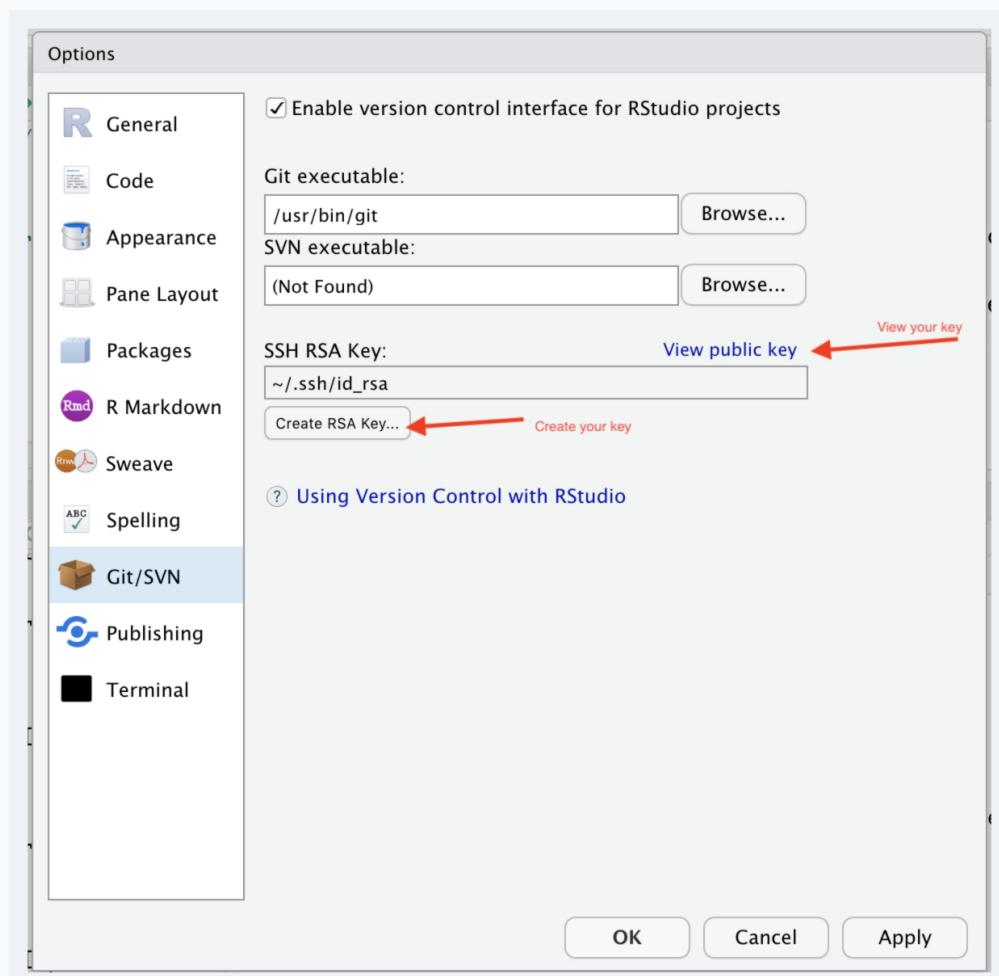
- Windows: C:/Program Files/Git/bin/git.exe
- Ubuntu: /usr/bin/git



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 'Personal settings' page under the 'Perfil' tab. The left sidebar lists various settings categories: Profile, Account, Account security, Security log, Security & analysis, Emails, Notifications, Scheduled reminders, Billing, SSH and GPG keys (which is highlighted with a red box), Repositories, and 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 Added on May 3, 2020 Last used within the last 2 weeks — Read/write	Delete
---	------------------------

[New SSH key](#)

GPG keys

There are no GPG keys associated with your account.

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

[New GPG key](#)

GitHub e RStudio

No GitHub

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

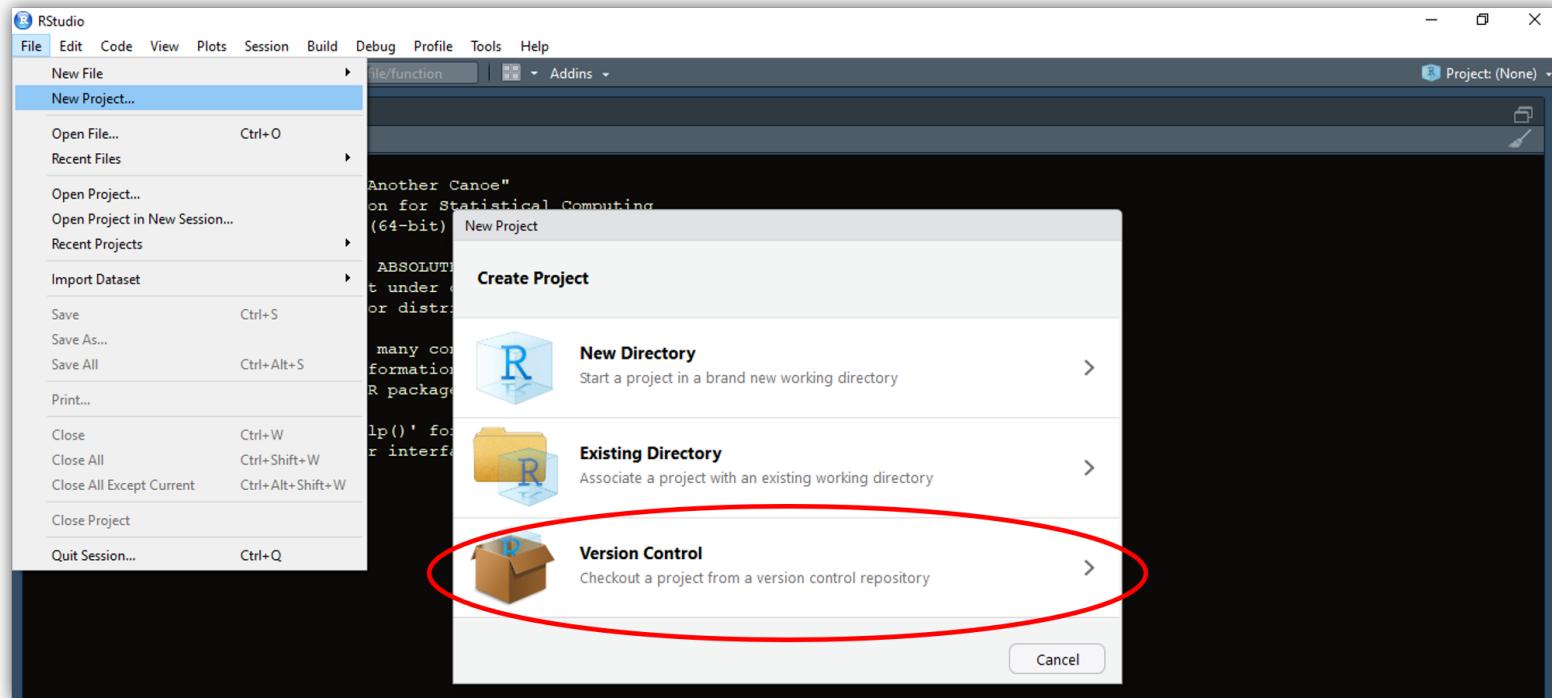
The screenshot shows the GitHub 'Personal settings' sidebar on the left with the 'SSH and GPG keys' option selected. The main area is titled 'SSH keys / Add new'. It contains two input fields: 'Title' and 'Key', both outlined with red boxes. Below the 'Key' field is a note: 'Begins with 'ssh-rsa', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521''. At the bottom is a green 'Add SSH key' button, also outlined with a red box.

Tudo certo até aqui?

1.7 Iniciando: init ou clone

Criar um Projeto R com controle de versão

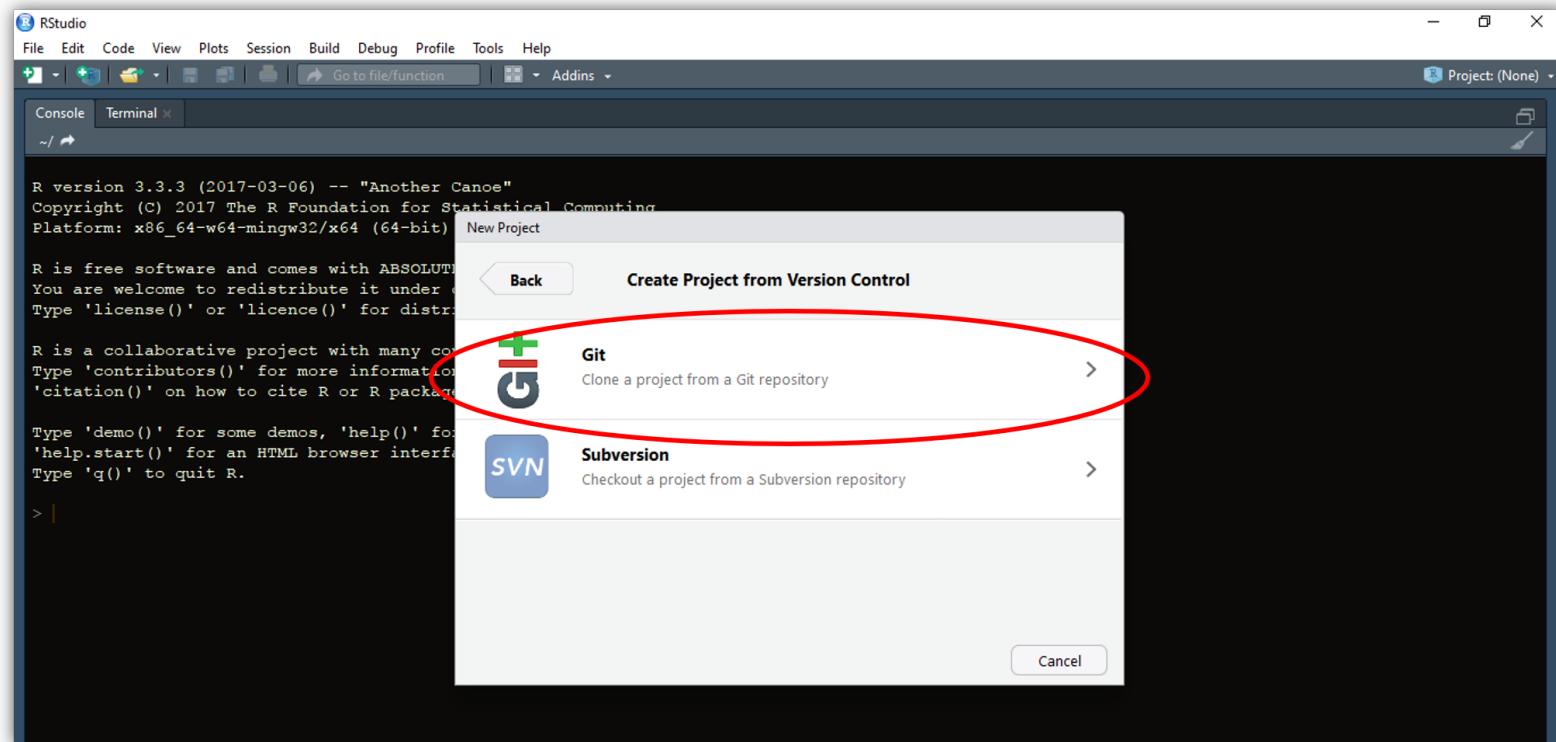
File > New Project > Version Control



1.7 Iniciando: init ou clone

Escolher clonar repositório do GitHub

Git

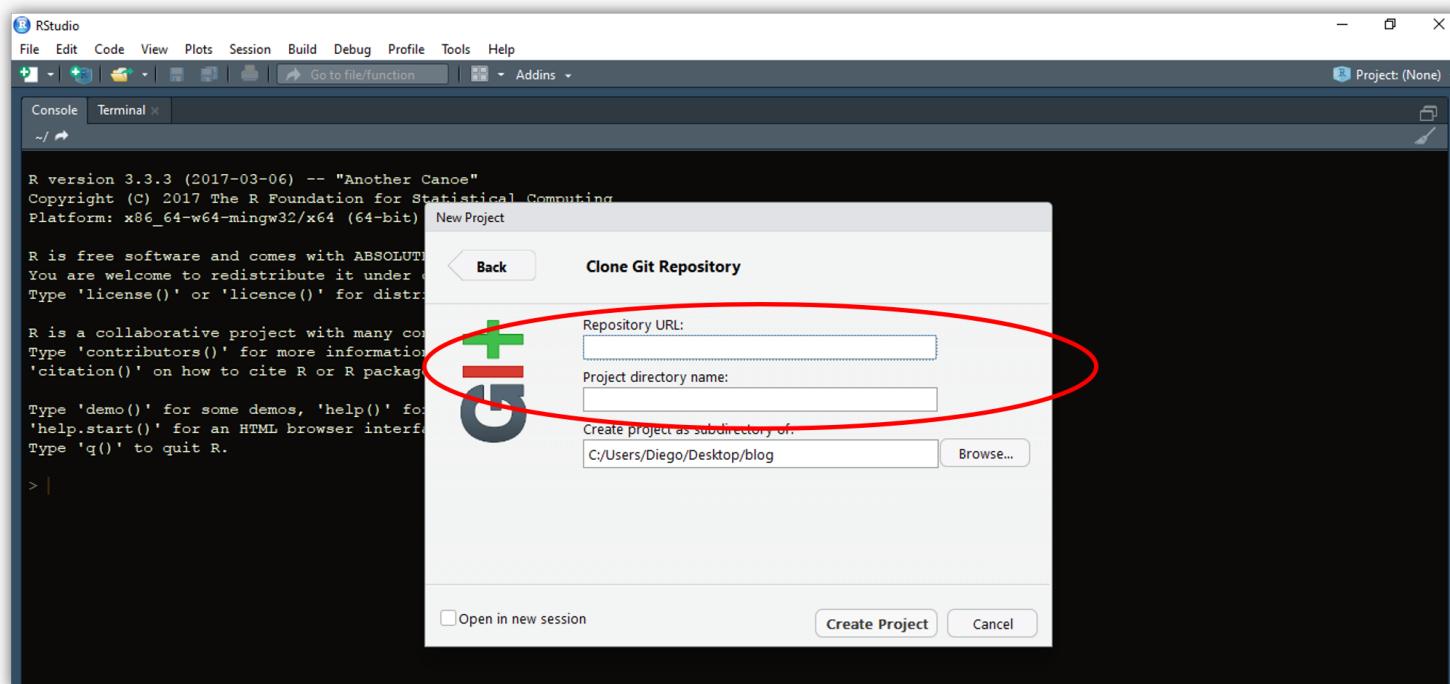


1.7 Iniciando: init ou clone

Preencher:

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

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



1.7 Iniciando: init ou clone

Terminal

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

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



Vamos aguardar o download

Agora vamos configurar o git

1.8 Configurando: config

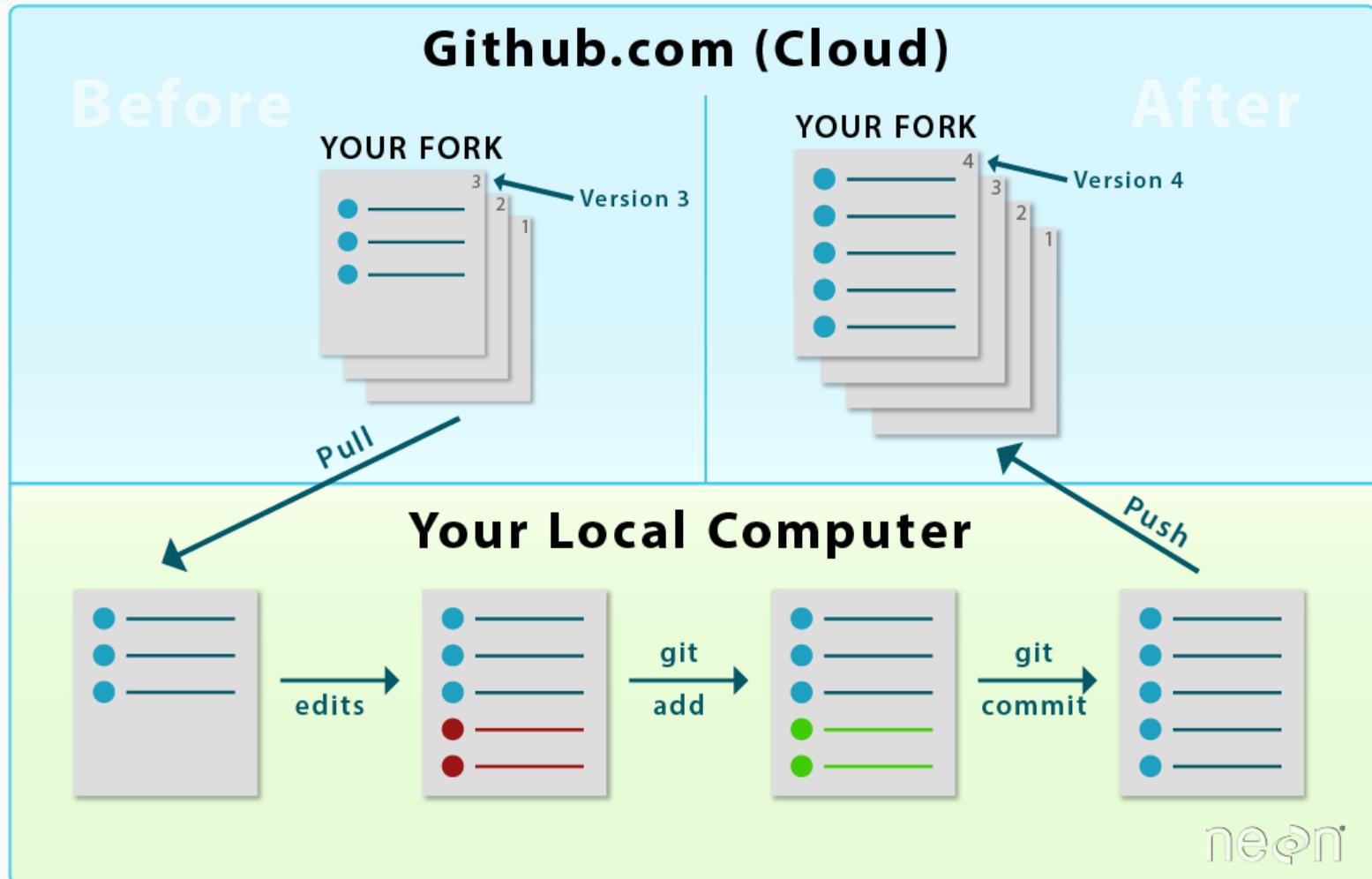
Precisamos **definir as configurações** do git

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



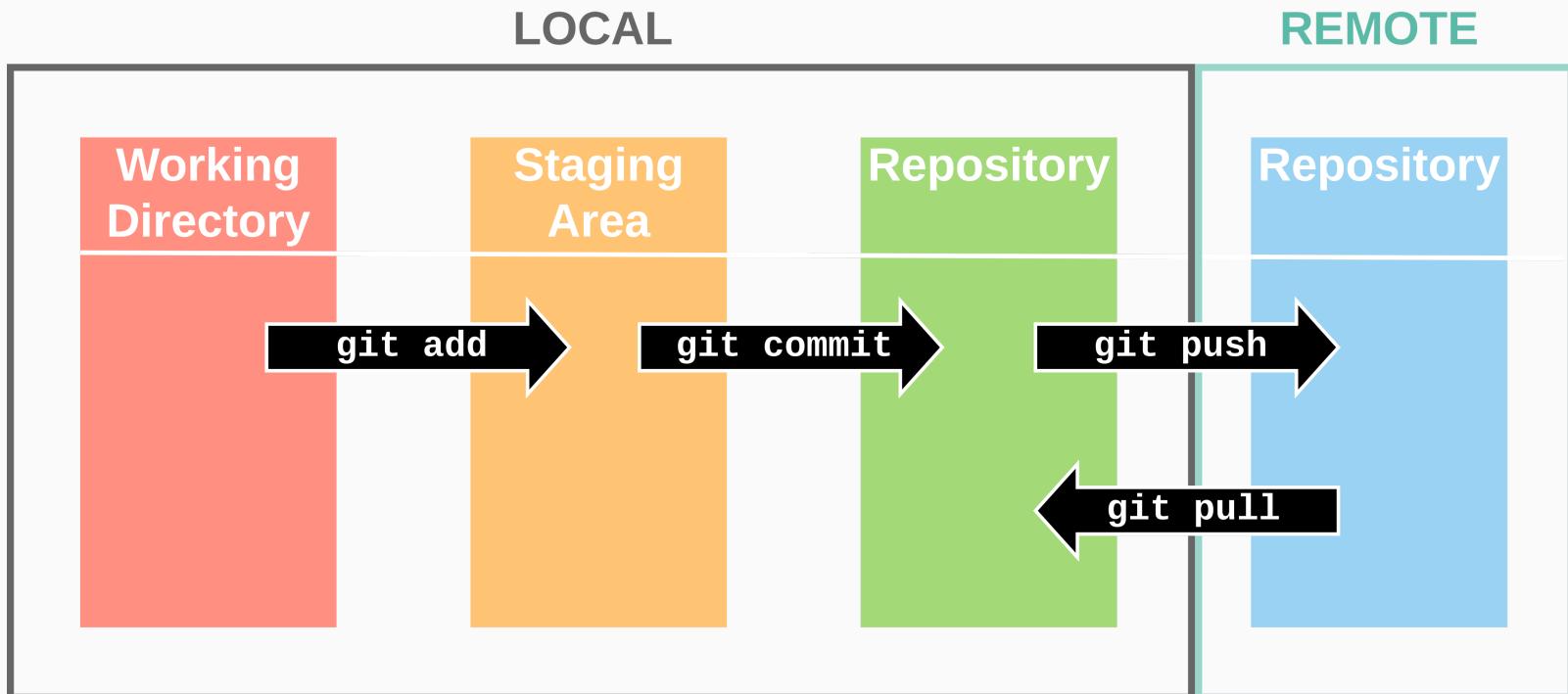
Git Config

Fluxo de trabalho com o git e GitHub



Fluxo de trabalho com o git e GitHub

Comandos para usar o git

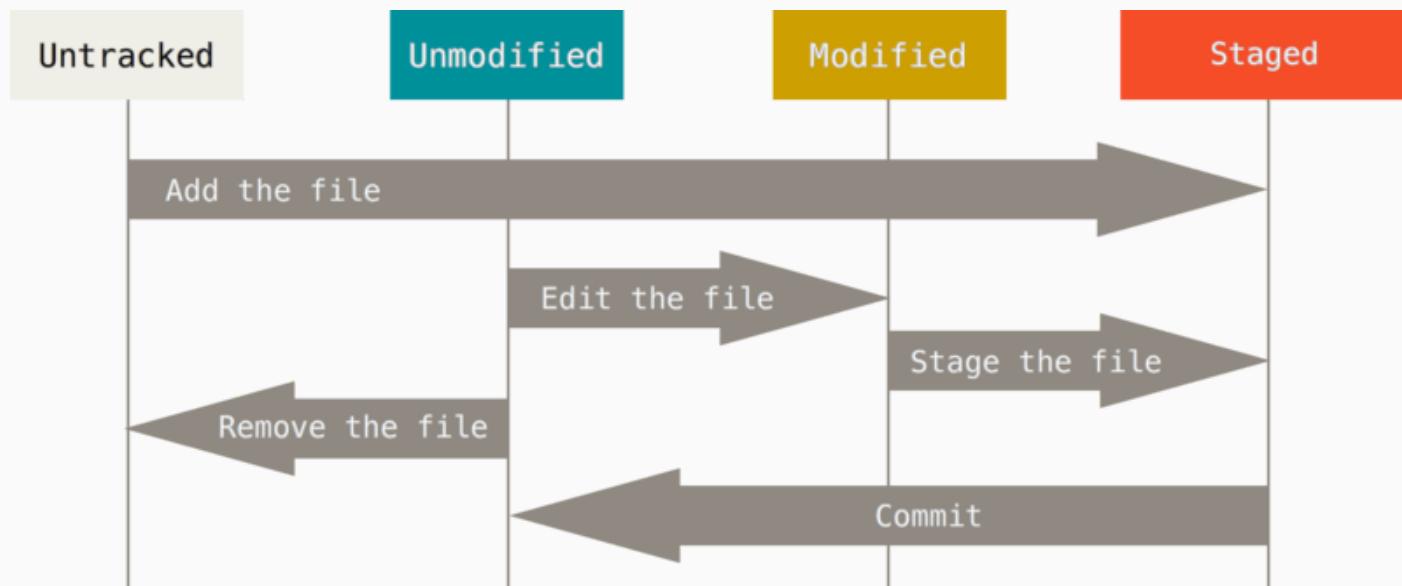


Agora prestem bastante atenção, porque
complica...

Git Status

Estados do repositório local do git:

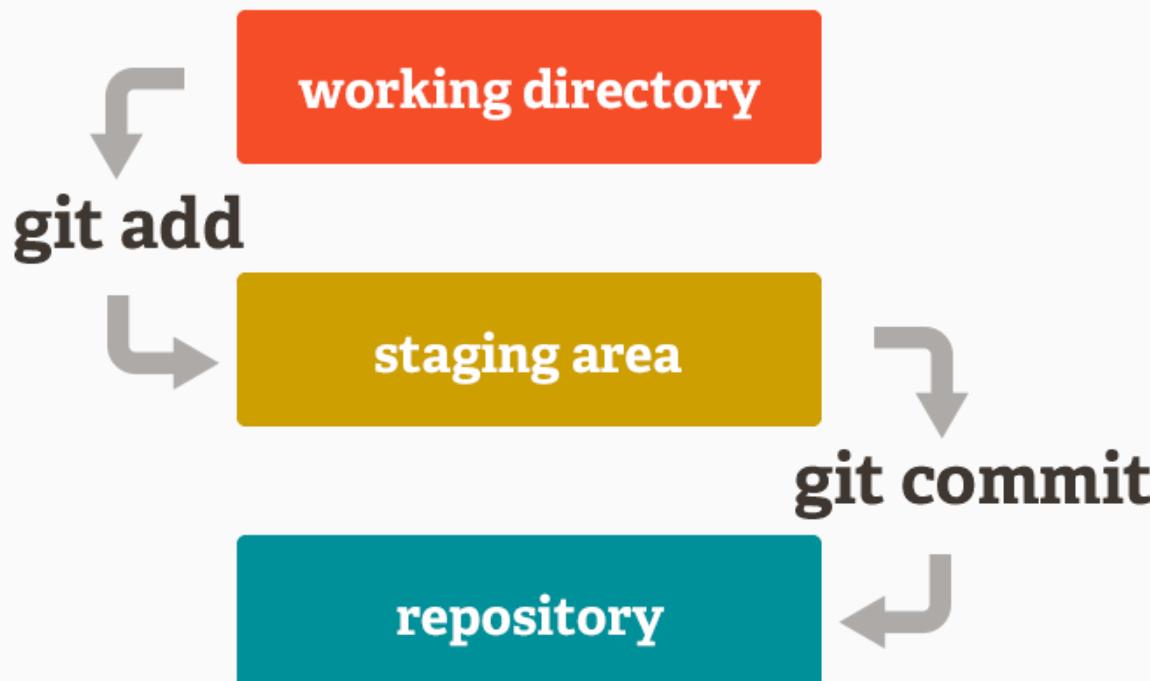
1. **Untracked**: não registrou o arquivo criado ou deletado
 2. **Unmodified**: marcou os arquivos como não-modificados após o commit
 3. **Modified**: marcou os arquivos modificados depois da edição
 4. **Staged**: marcou o arquivo adicionado ou modificado para ir para o commit
- **Commit**: os arquivos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



Git Sections

Seções de um repositório git:

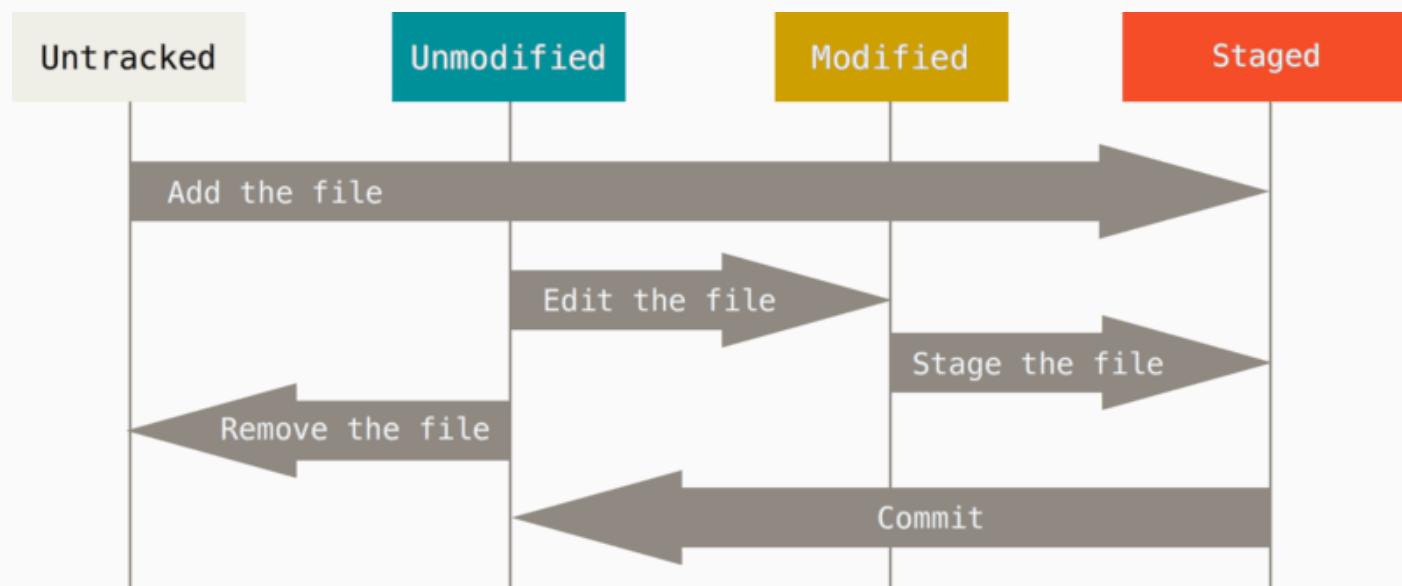
1. **Working directory**: arquivos não-trackeados
2. **Staging area**: arquivos trackeados (adicionados ou modificados)
3. **Repository**: arquivos comitados (.git)



1.9 Básico: status, add, commit e log

status: mostra o status do repositório

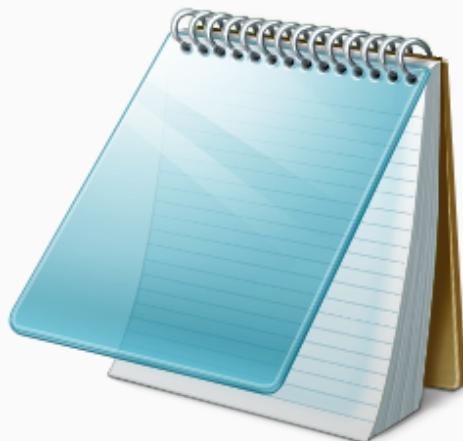
```
git status
```



Vamos criar um arquivo no diretório

Usando a aba **Terminal** do RStudio

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

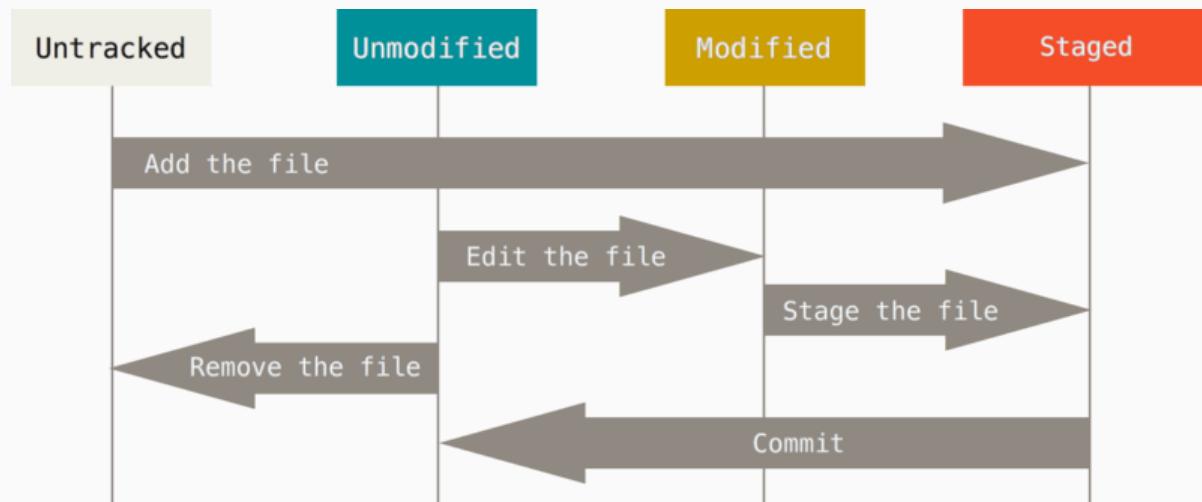


1.9 Básico: status, add, commit e log

status: mostra o status do repositório

```
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")
```

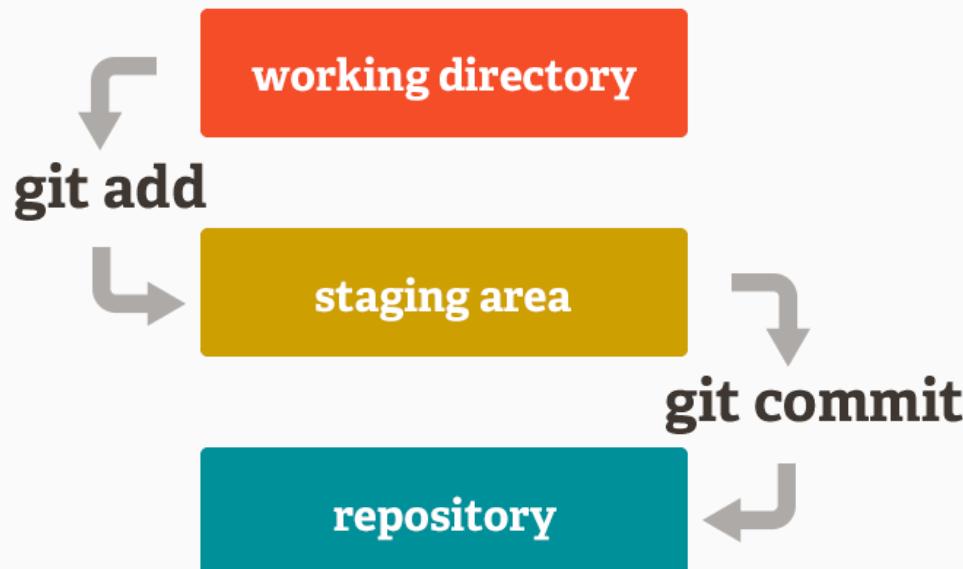


1.9 Básico: status, add, commit e log

add: adiciona mudanças após edições (staging area)

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

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

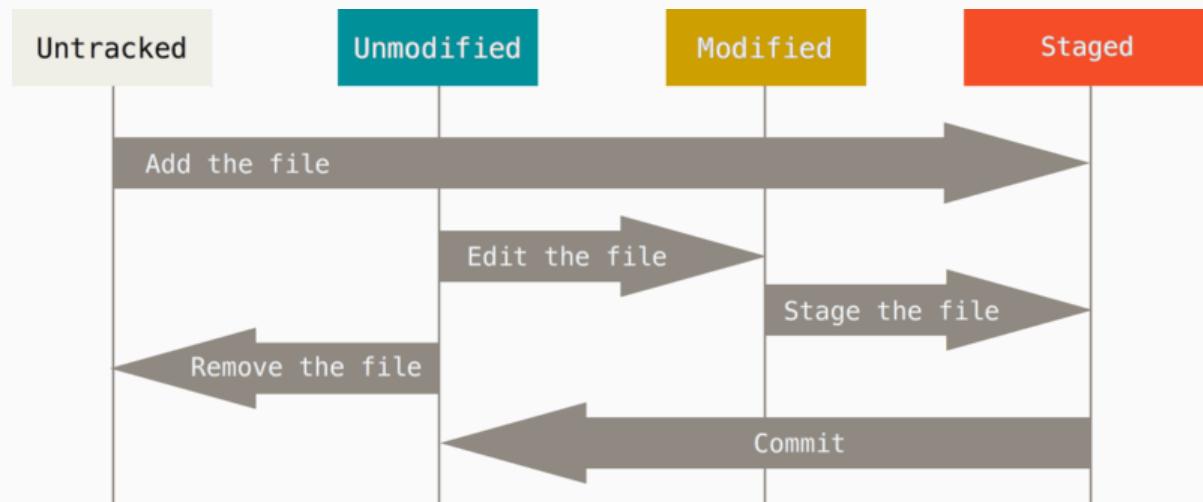


1.9 Básico: status, add, commit e log

status: mostra o status do repositório

```
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

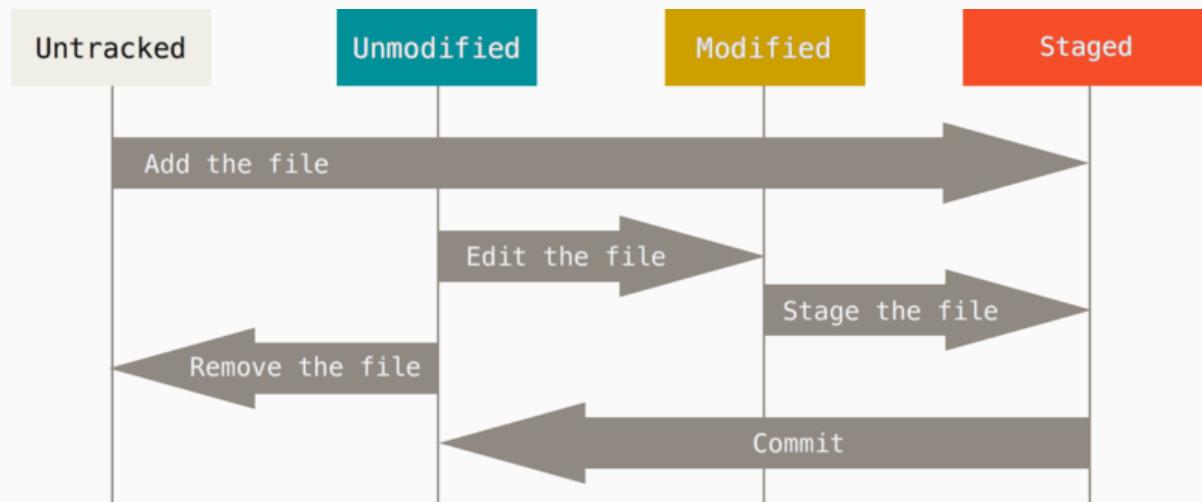


1.9 Básico: status, add, commit e log

status: mostra o status do repositório

```
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
```

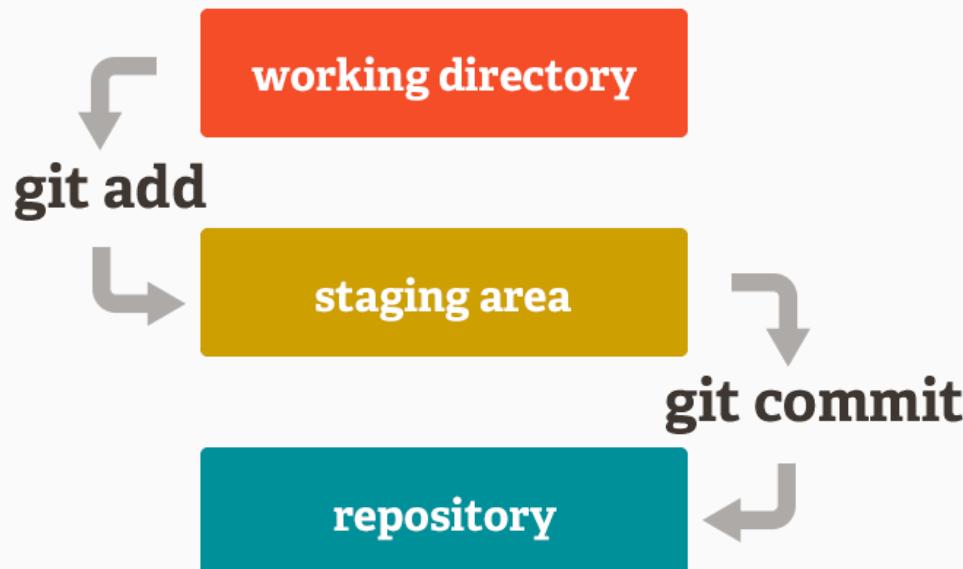


1.9 Básico: status, add, commit e log

add: adiciona mudanças após edições (staging area)

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

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

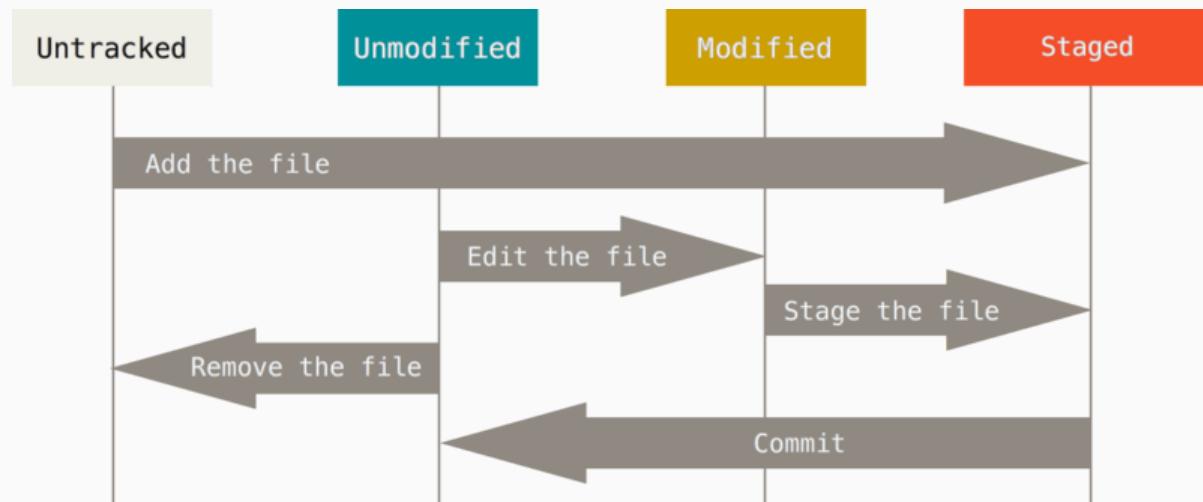


1.9 Básico: status, add, commit e log

status: mostra o status do repositório

```
git status
```

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

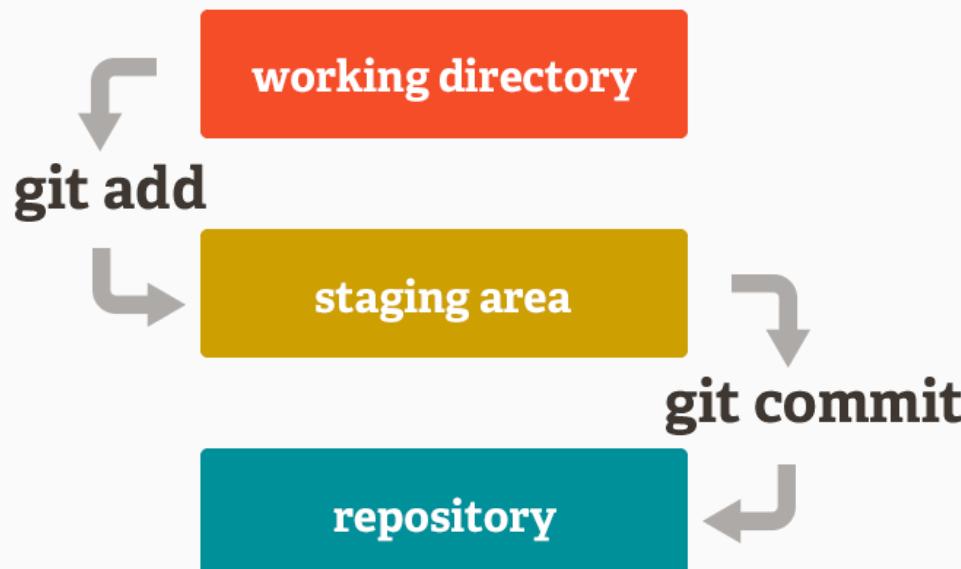


1.9 Básico: status, add, commit e log

commit: armazena as mudanças com uma descrição, nova 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!

- As mensagens no commit são fundamentais!
- Devem ser curtas e indicar as mudanças!

	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	ADKFJSLKDFJSDFKLJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

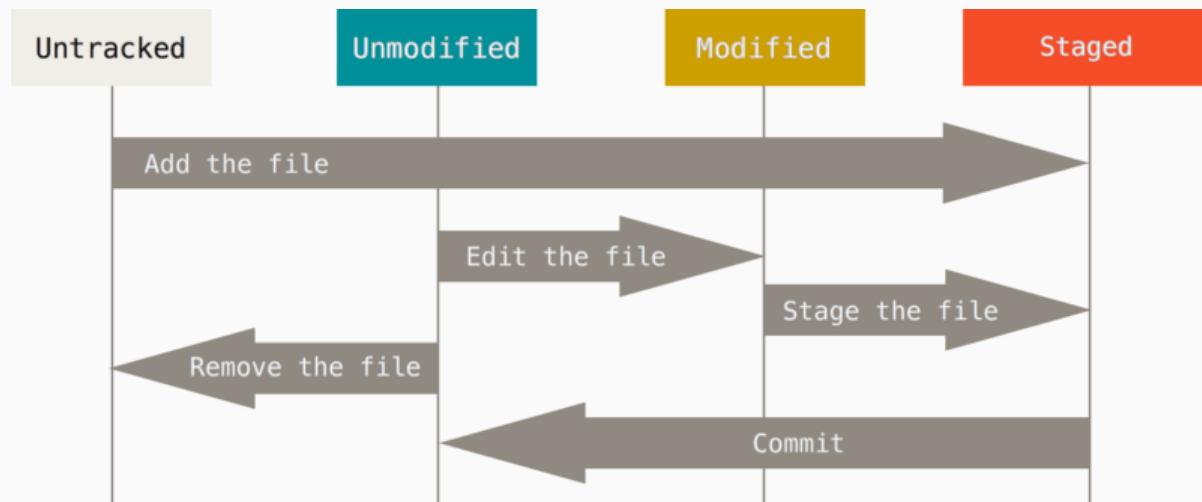
AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

1.9 Básico: status, add, commit e log

status: mostra o estatus do repositório

```
git status
```

```
## On branch master
## Your branch is ahead of 'origin/master' by 1 commit.
##   (use "git push" to publish your local commits)
##
## nothing to commit, working tree clean
```



1.9 Básico: status, add, commit e log

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

```
git log
```

```
## commit 3ae6ecec901c8abbf39563a995978a78abe5919 (HEAD → master)
## Author: mauriciovancine <mauricio.vancine@gmail.com>
## Date:   Sun Oct 18 19:06:34 2020 -0300
##
##       2020-10-19 add teste.txt
```

1.9 Básico: status, add, commit e log

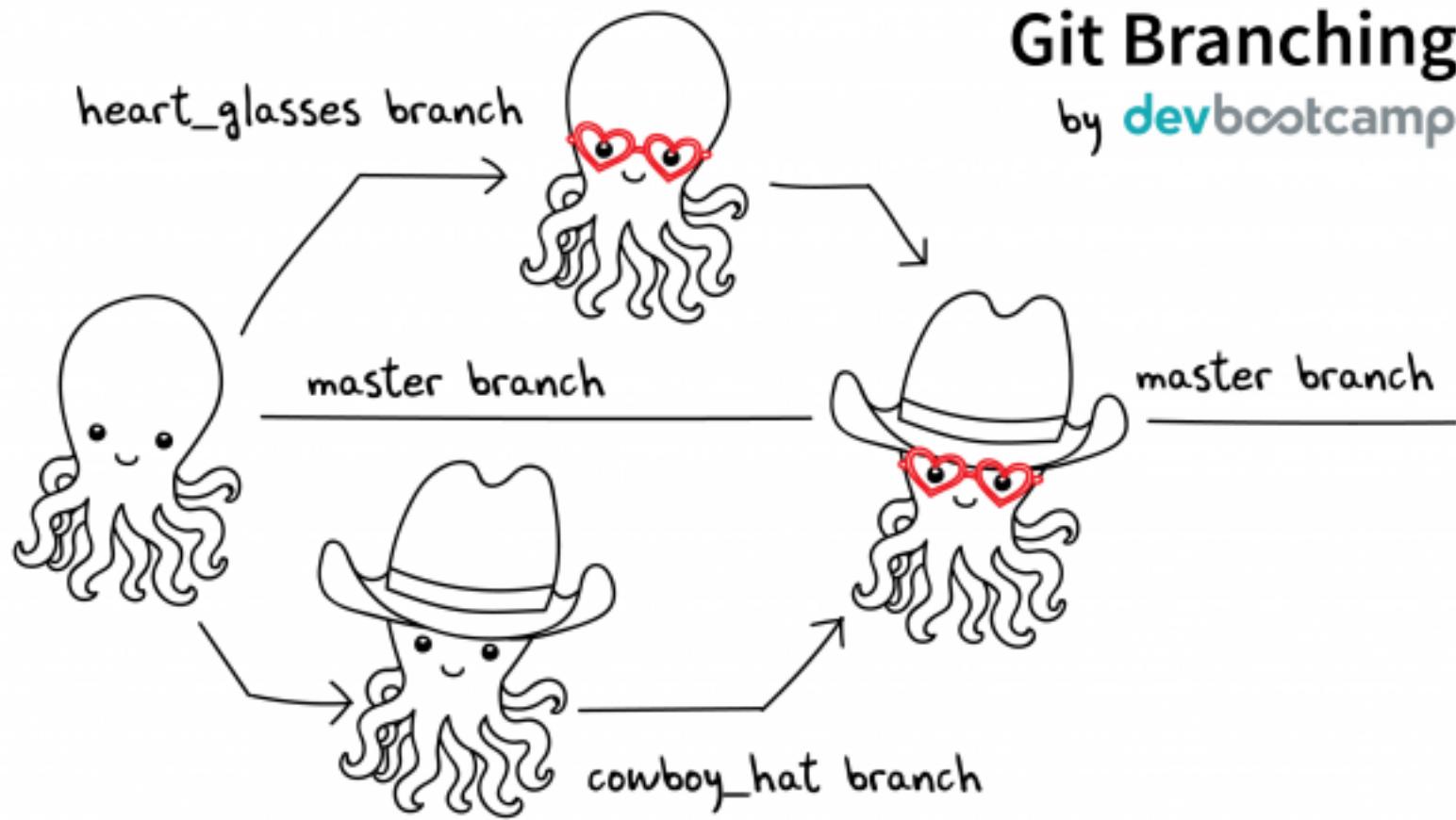
log: registro de todo o histórico de commits (tags)

```
git log --oneline
```

```
## 3ae6ece (HEAD → master) 2020-10-19 add teste.txt
## 18f482f (origin/master, origin/HEAD) 2020-10-18
## dd8fe74 2020-10-18
## bbb5496 2020-10-17
## 4247eaa 2020-10-16
## 8fb3106 2020-10-15
## 0cb95a1 2020-10-15
## 22a9044 2020-10-13
## c4f0201 2020-10-13
## 867a1a5 2020-10-13
## 297e808 2020-10-13
## 68c310a 2020-10-12
## ...
```

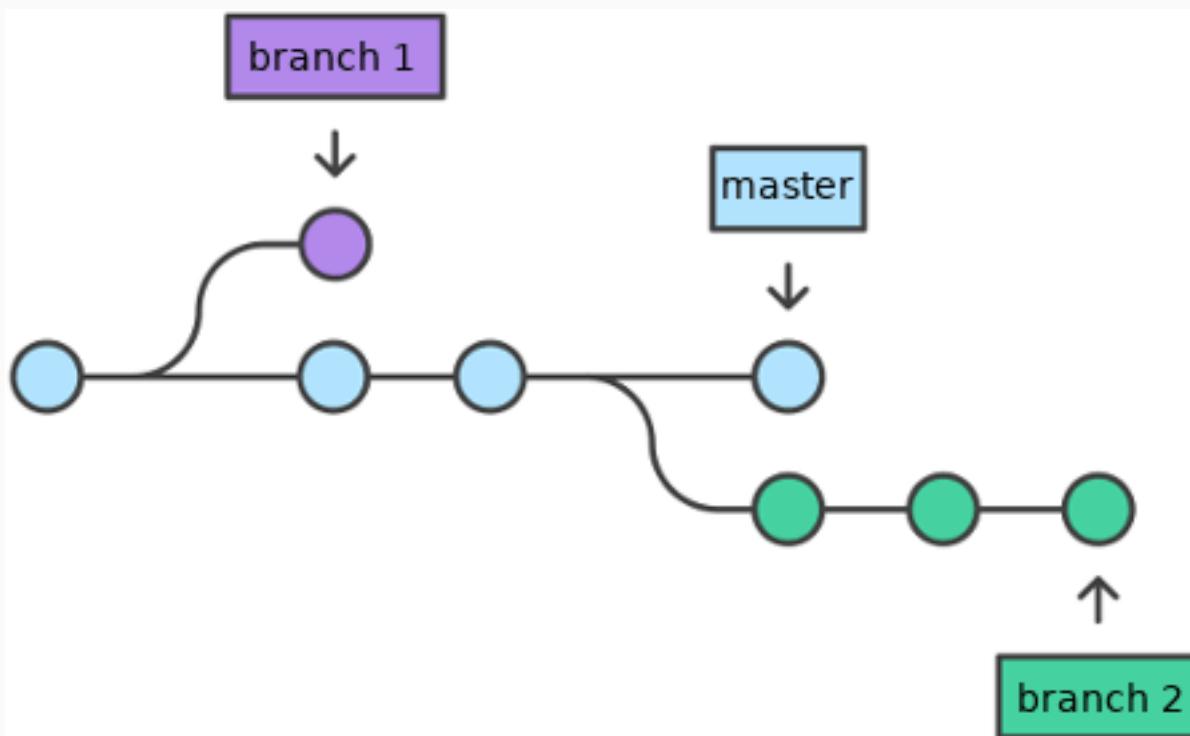
Git Branching

by **devbootcamp**



1.10 Ramificações

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

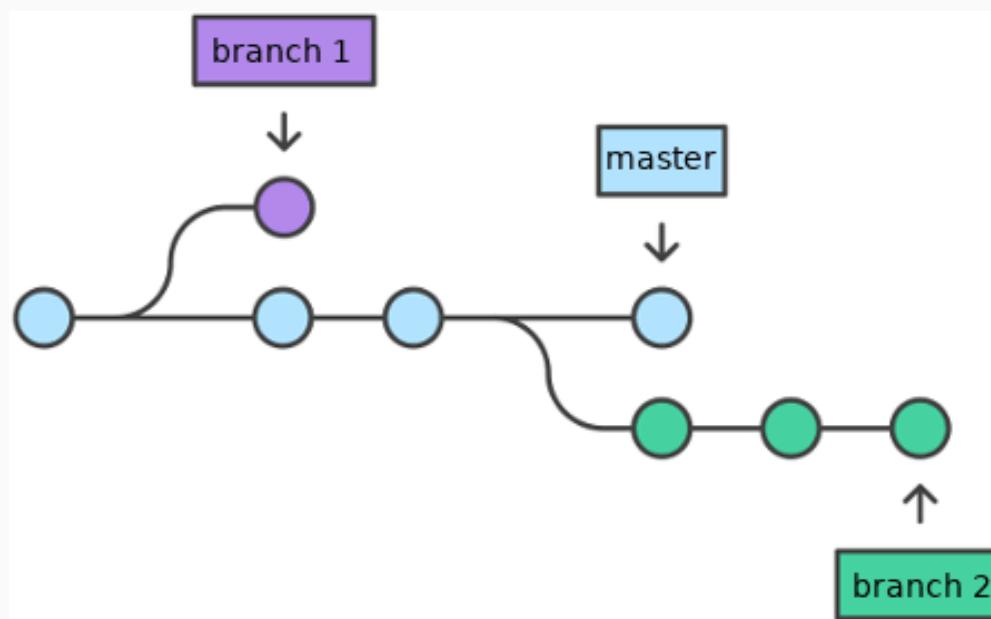


1.10 Ramificações

branch: listar os branches e **verificar** o branch de edição

```
git branch
```

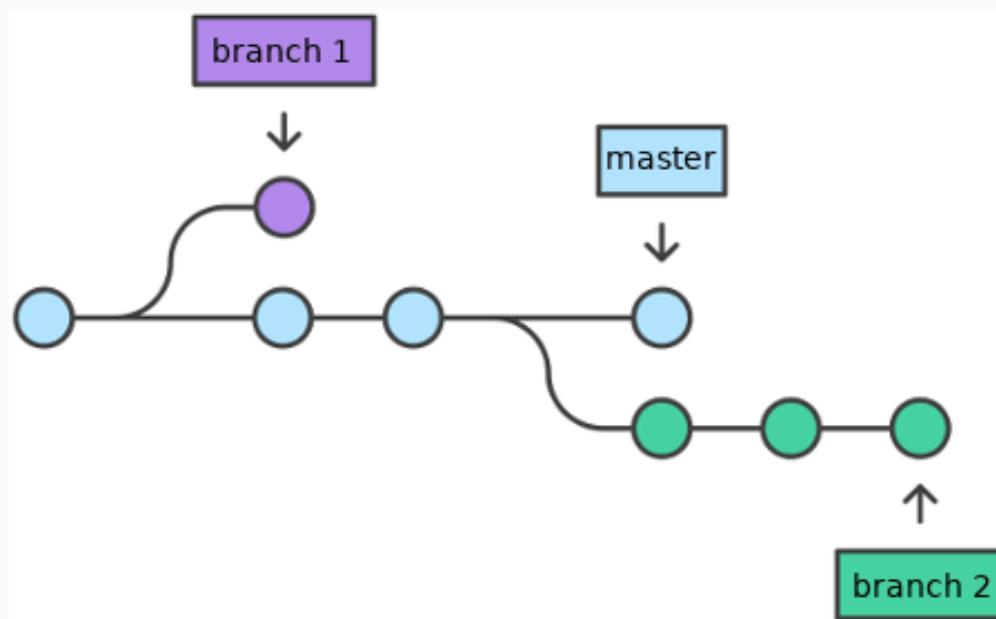
```
## * master
```



1.10 Ramificações

branch: criar uma ramificação

```
git branch branch-1
```

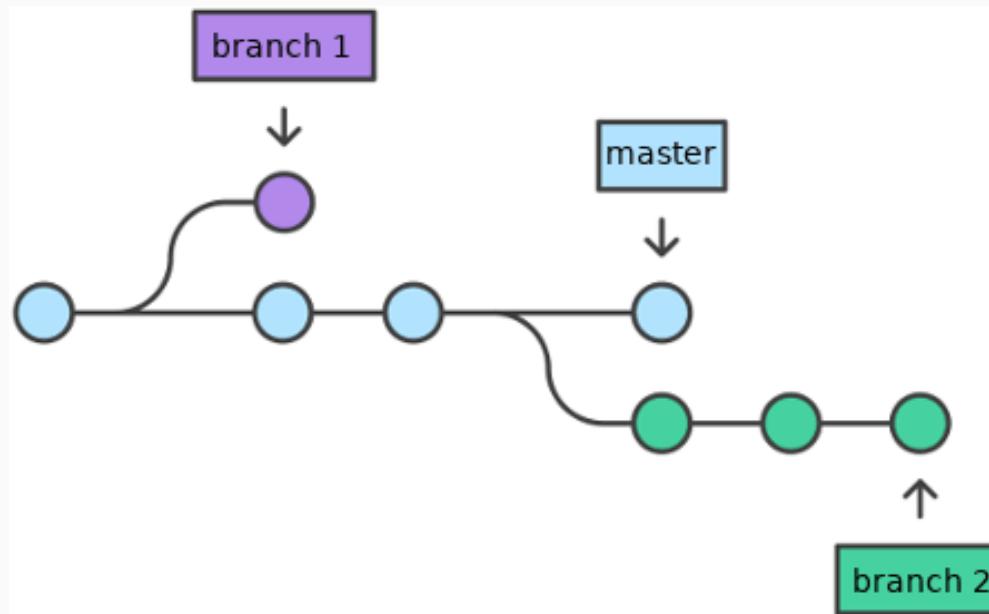


1.10 Ramificações

branch: listar os branches e **verificar** o branch de edição

```
git branch
```

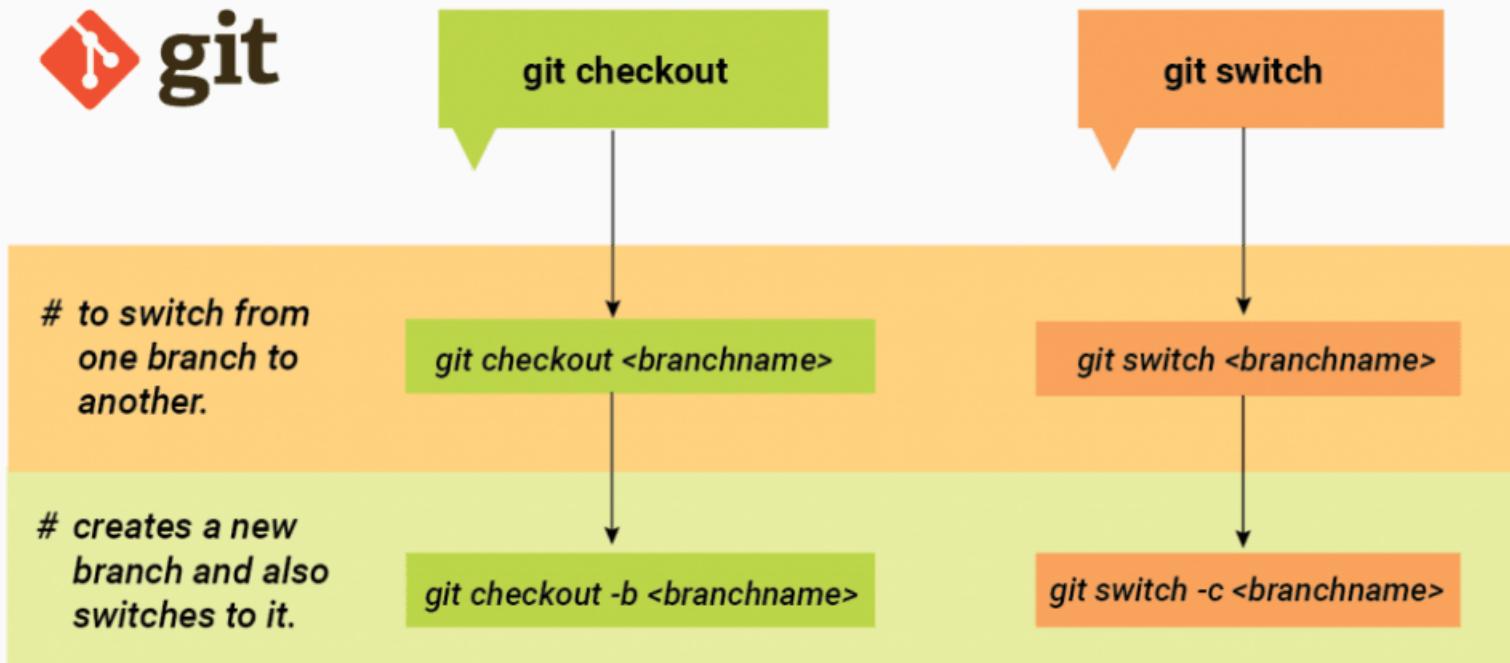
```
## branch-1  
## * master
```



1.10 Ramificações

switch: troca a raficação de edição

```
git switch branch-1
```

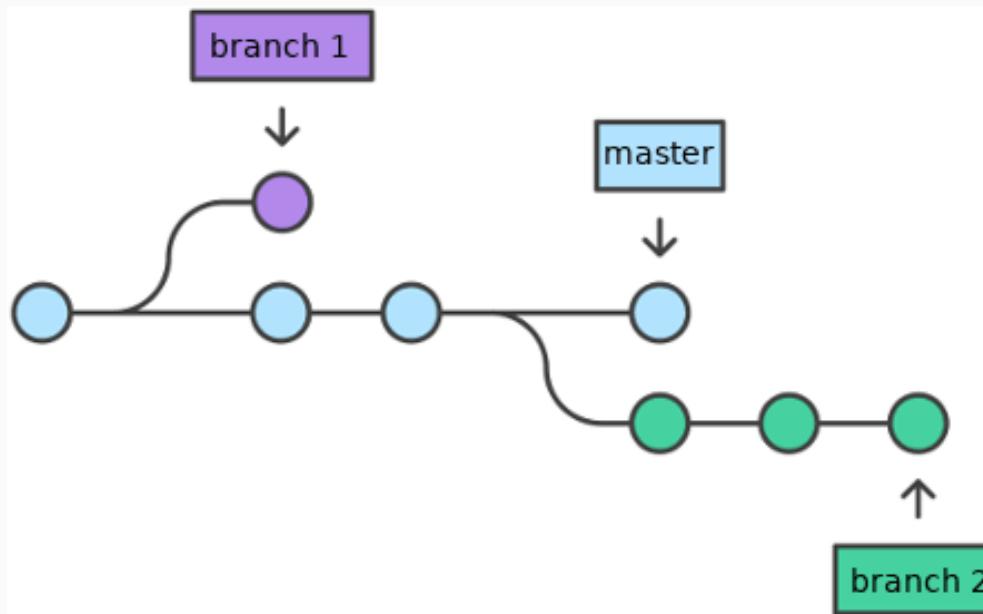


1.10 Ramificações

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

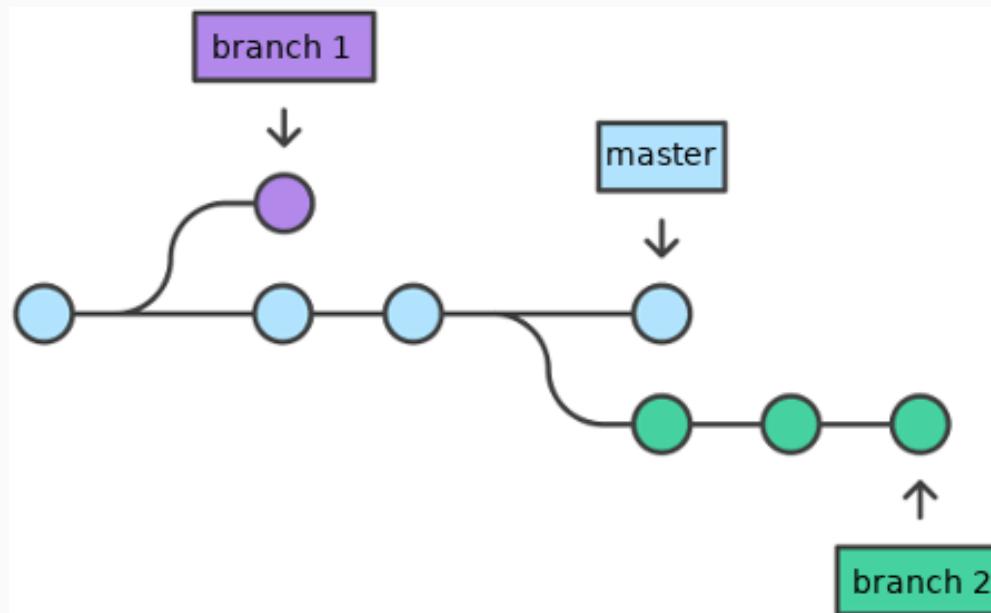
```
# terminal  
touch teste_branch1.txt  
git add -Av  
git commit -m "2020-10-19"
```



1.10 Ramificações

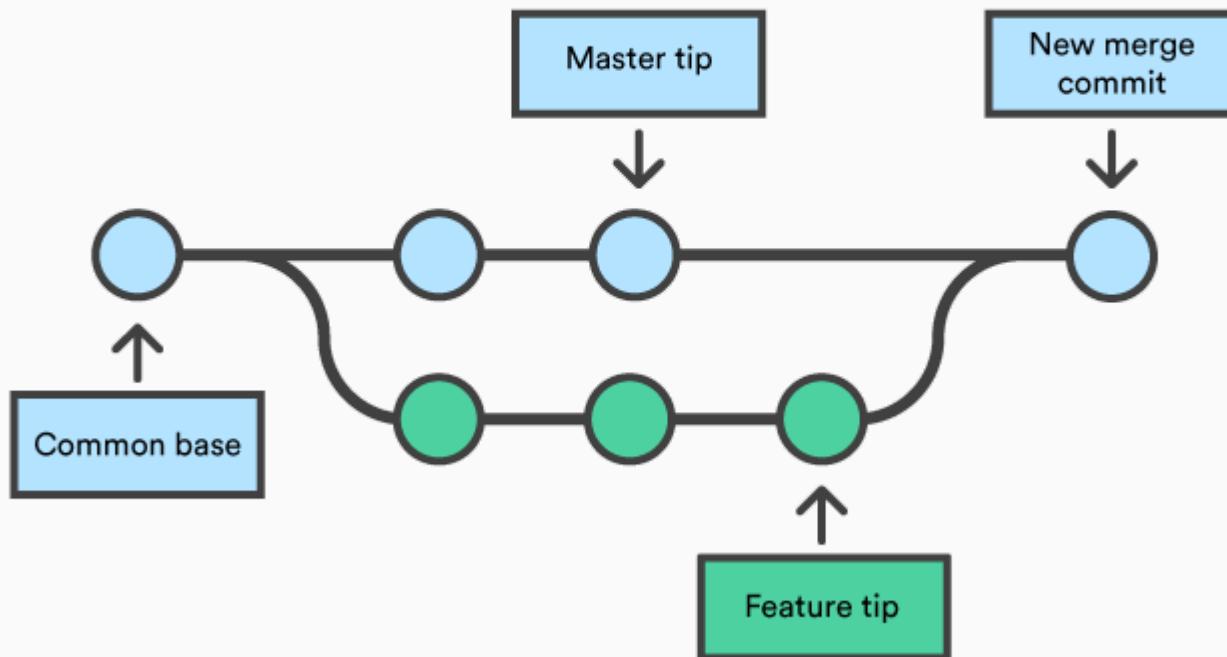
switch: troca a raficação de edição

```
git switch master
```



1.10 Ramificações

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

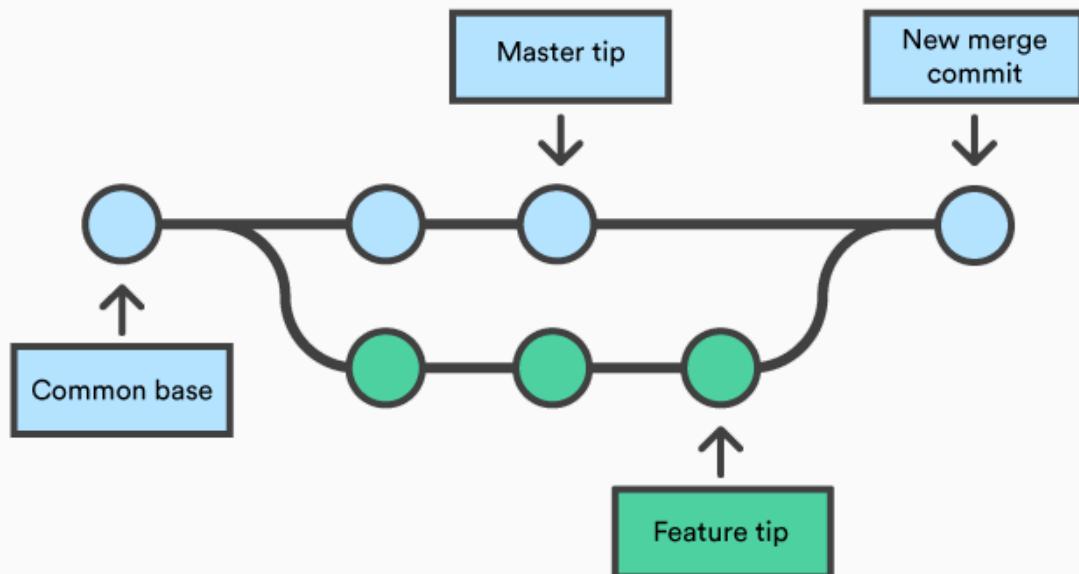


1.10 Ramificações

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

```
git merge branch-1
```

```
## teste_branch1.txt / 0
```



1.10 Ramificações

CUIDADO!



1.10 Ramificações

CUIDADO!



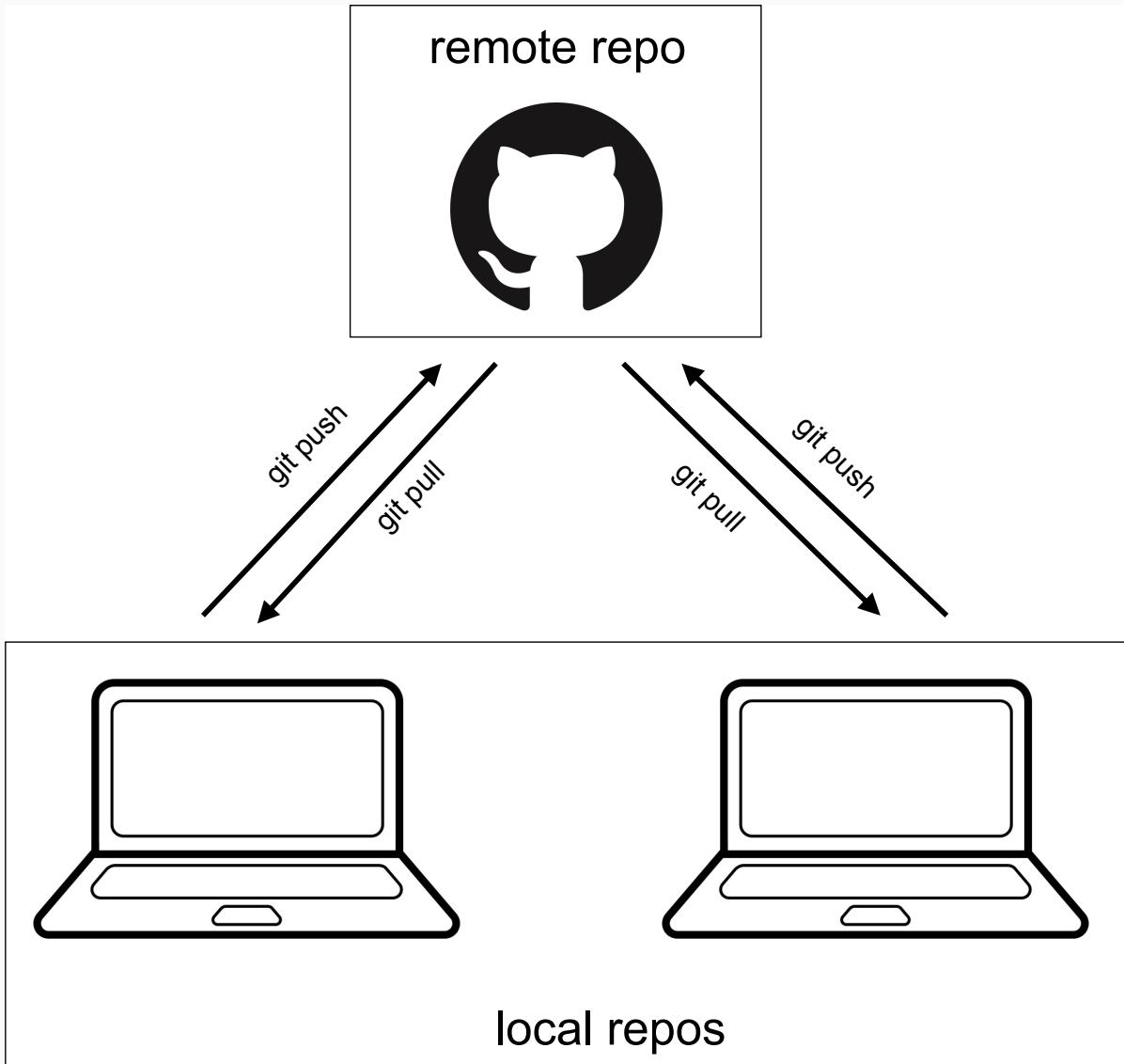
Origin

Master

>git merge

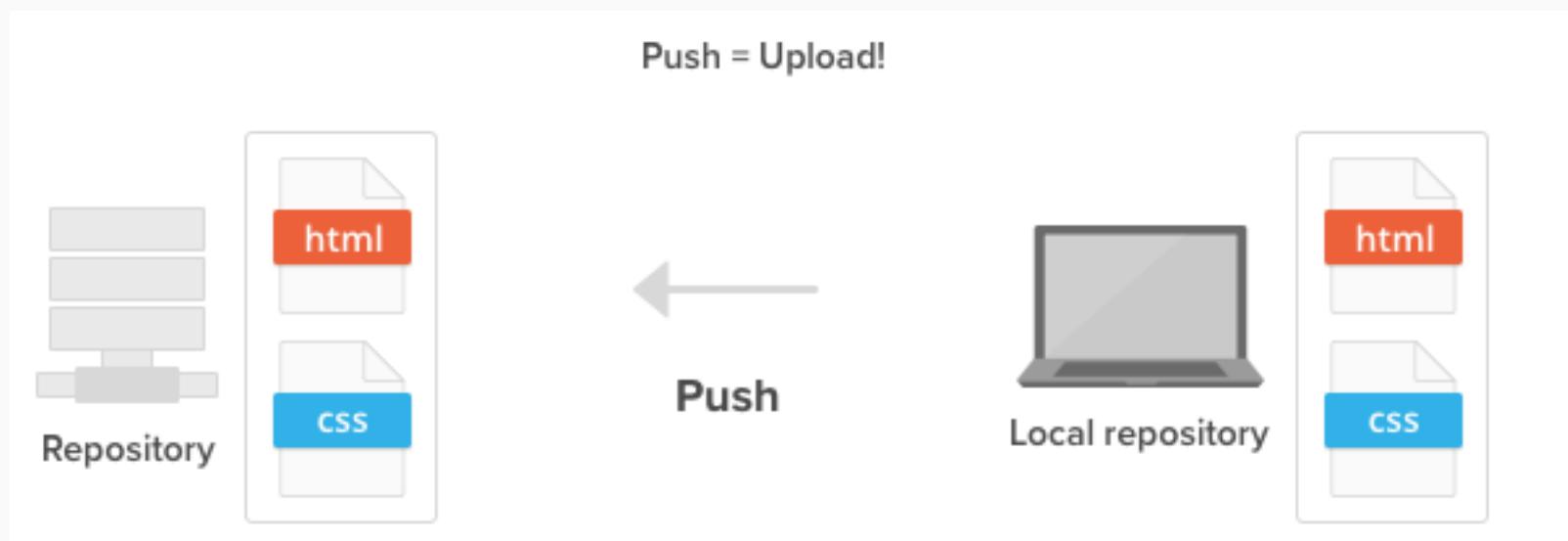


Merging would be like...



1.11 Remoto: push e pull

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



1.11 Remoto: push e pull

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

The diagram illustrates the components of the `git push -u origin master` command. It features a central command line with annotations pointing to specific words:

- An arrow points to the word `-u` with the label "sets upstream tracking branch".
- The word `origin` is bracketed with a wavy line and labeled "remote repo alias".
- The word `master` is bracketed with a wavy line and labeled "branch".
- The word `git` is bracketed with a wavy line and labeled "primary command to push commits to remote repo".
- The word `push` is bracketed with a wavy line and labeled "alias of remote repo".

1.11 Remoto: push e pull

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

```
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

Em caso de incêndio...

In case of fire



1. git commit



2. git push



3. leave building

1.11 Remoto: push e pull

CUIDADO!



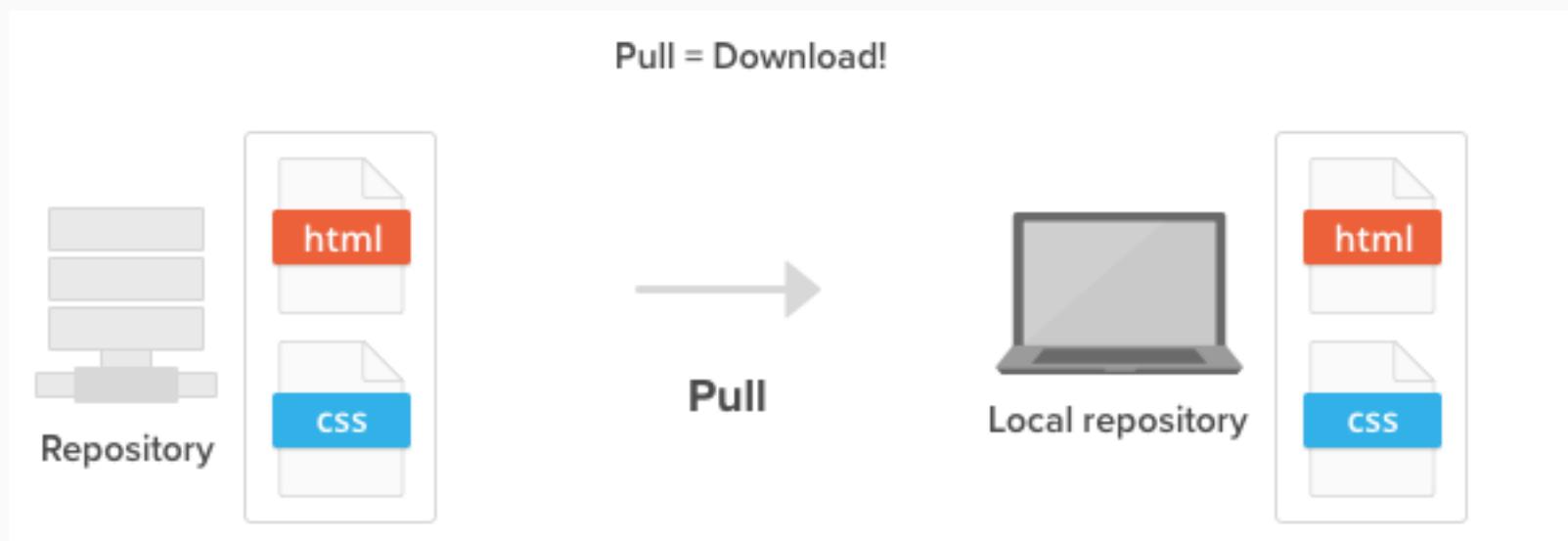
1.11 Remoto: push e pull

CUIDADO!



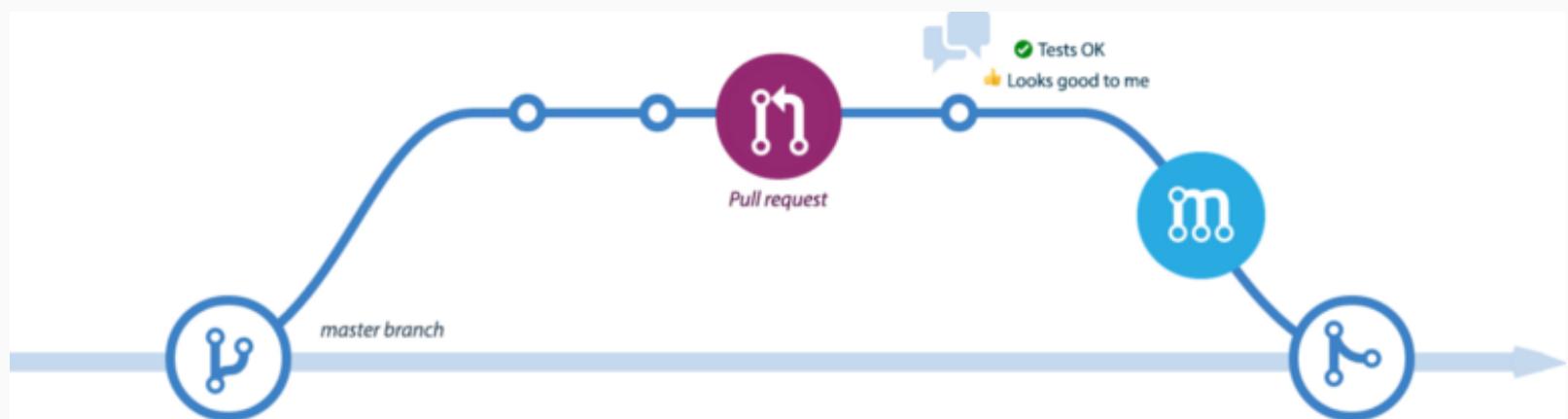
1.11 Remoto: push e pull

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**



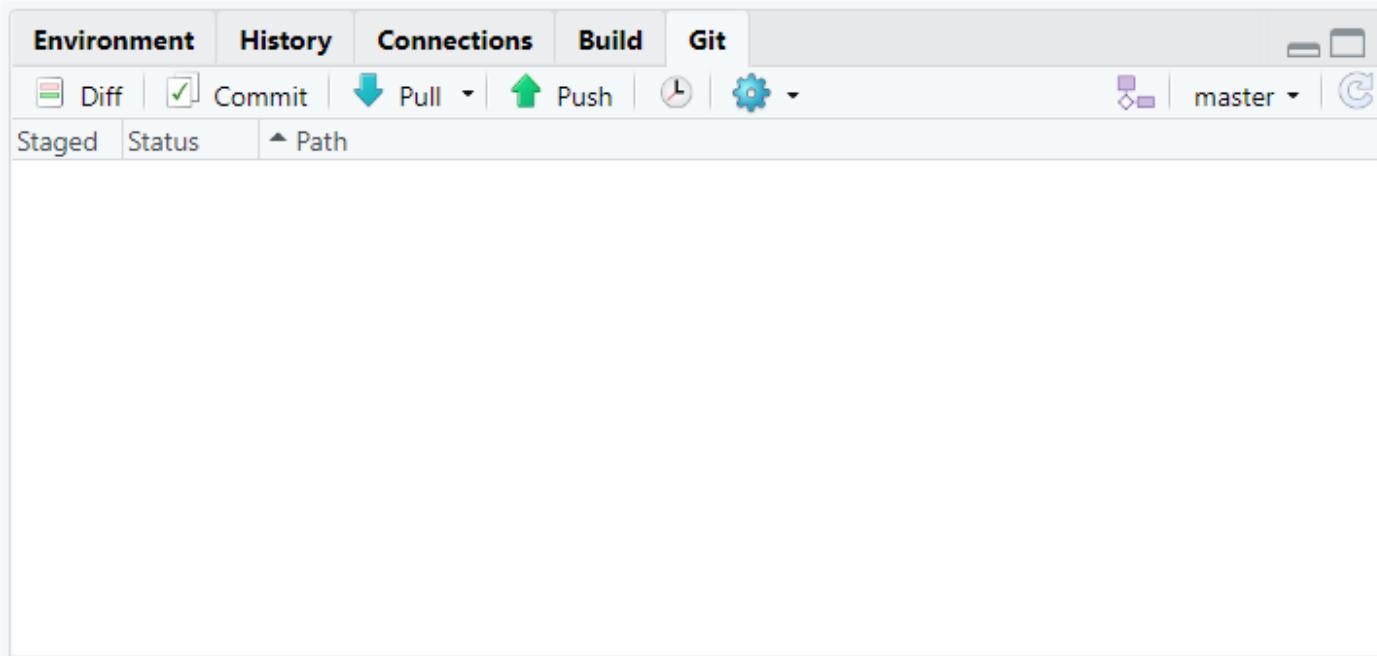
Interface Gráfica RStudio

Agradecimento à Beatriz Milz pelas
figuras

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



Stage files:

A

Added

D

Deleted

M

Modified

R

Renamed

?

Untracked

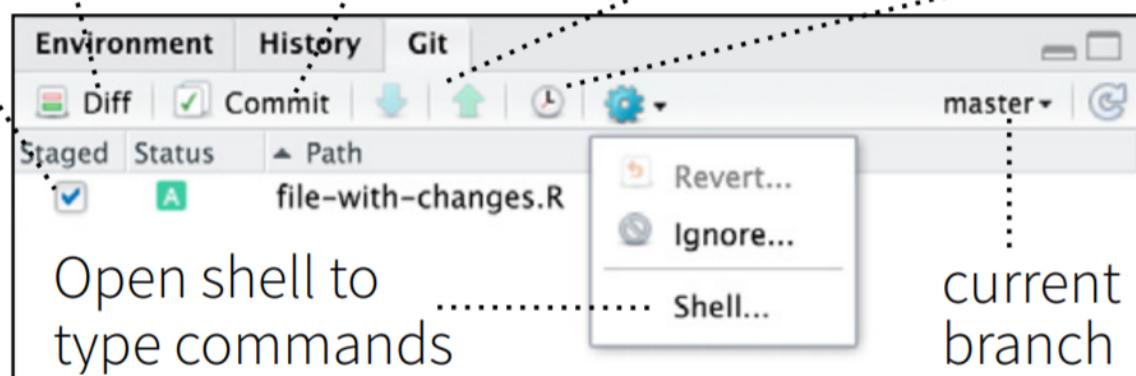
Turn on at **Tools > Project Options > Git/SVN**

Show file diff

Commit staged files

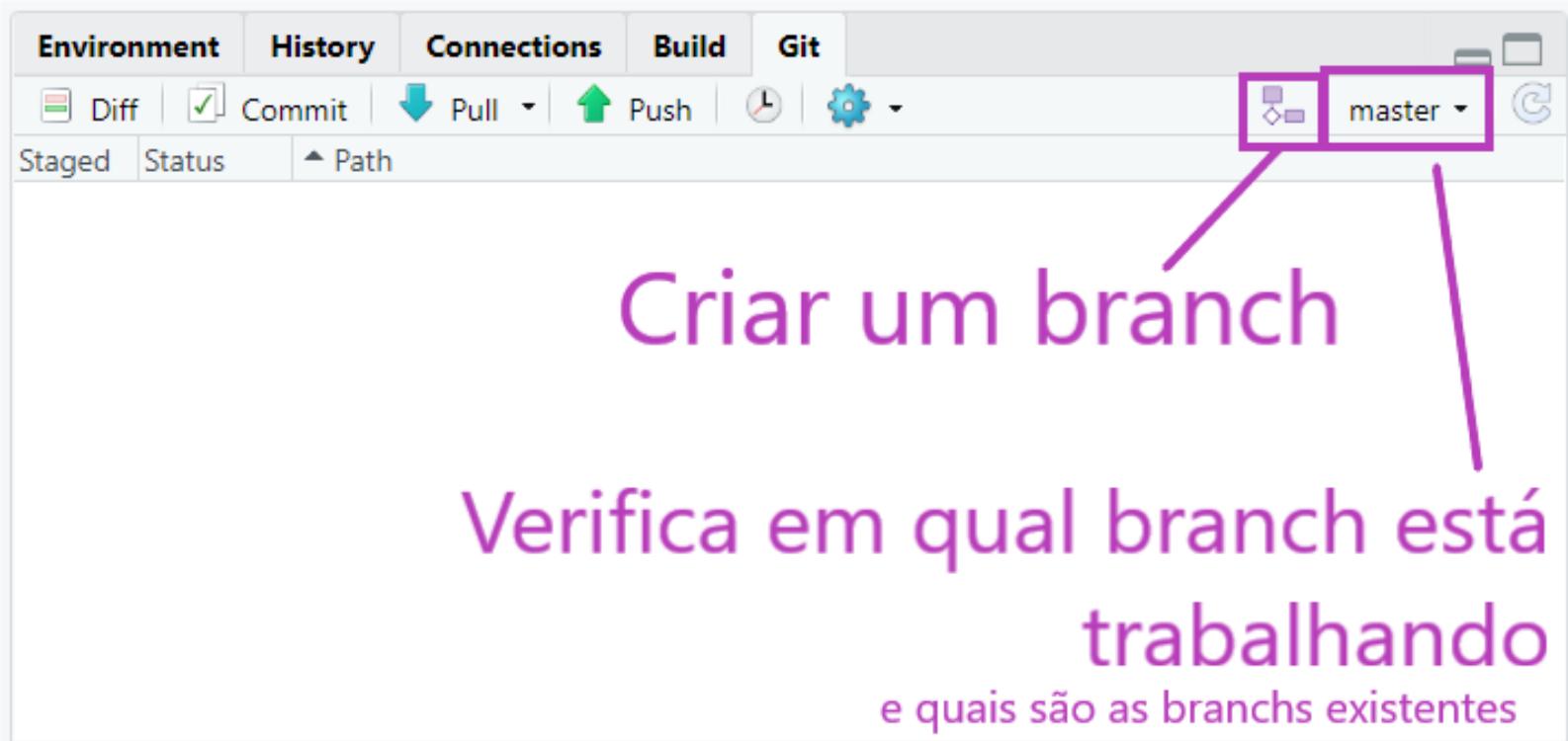
Push/Pull to remote

View History



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 panel has tabs for "Changes", "History", and "master". The "Changes" tab is selected, showing a list of files: "docs/index.Rmd" (M), "docs/index.html" (M), and "docs/img/prints/git-pane.PNG" (D). Below this is a "Commit message" field and buttons for "Amend previous commit" and "Commit". At the bottom, there are buttons for "Stage chunk" and "Discard chunk". The main area displays a diff of the "docs/index.Rmd" file:

```
@@ -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 usethis::create_from_github("rstudio-education/remaster-the-tidyverse", #usuário/repositório
537 usethis::create_from_github("rstudio-education/datascience-box", #usuário/repositório
538 538 destdir = "C:/Users/beatr/Documents/GitHub", #diretório onde quer que os arquivos sejam salvos
539 539 fork = TRUE)
540 # creating 'C:/Users/beatr/Documents/GitHub/remaster-the-tidyverse/'
541 # ✓ Forking 'rstudio-education/remaster-the-tidyverse'
542 # Which git protocol to use? (enter 0 to exit)
543 #
544 # 1: ssh  <-- presumes that you have set up ssh keys
545 {{# 2: https <-- choose this if you don't have ssh keys (or don't know if you do)}}
546 #
547 # Selection: 2
548 # • Tip: To suppress this menu in future, put
549 #   `options(usethis.protocol = "https")`
550 #   in your script or in a user- or project-level startup file, '.Rprofile'.
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:

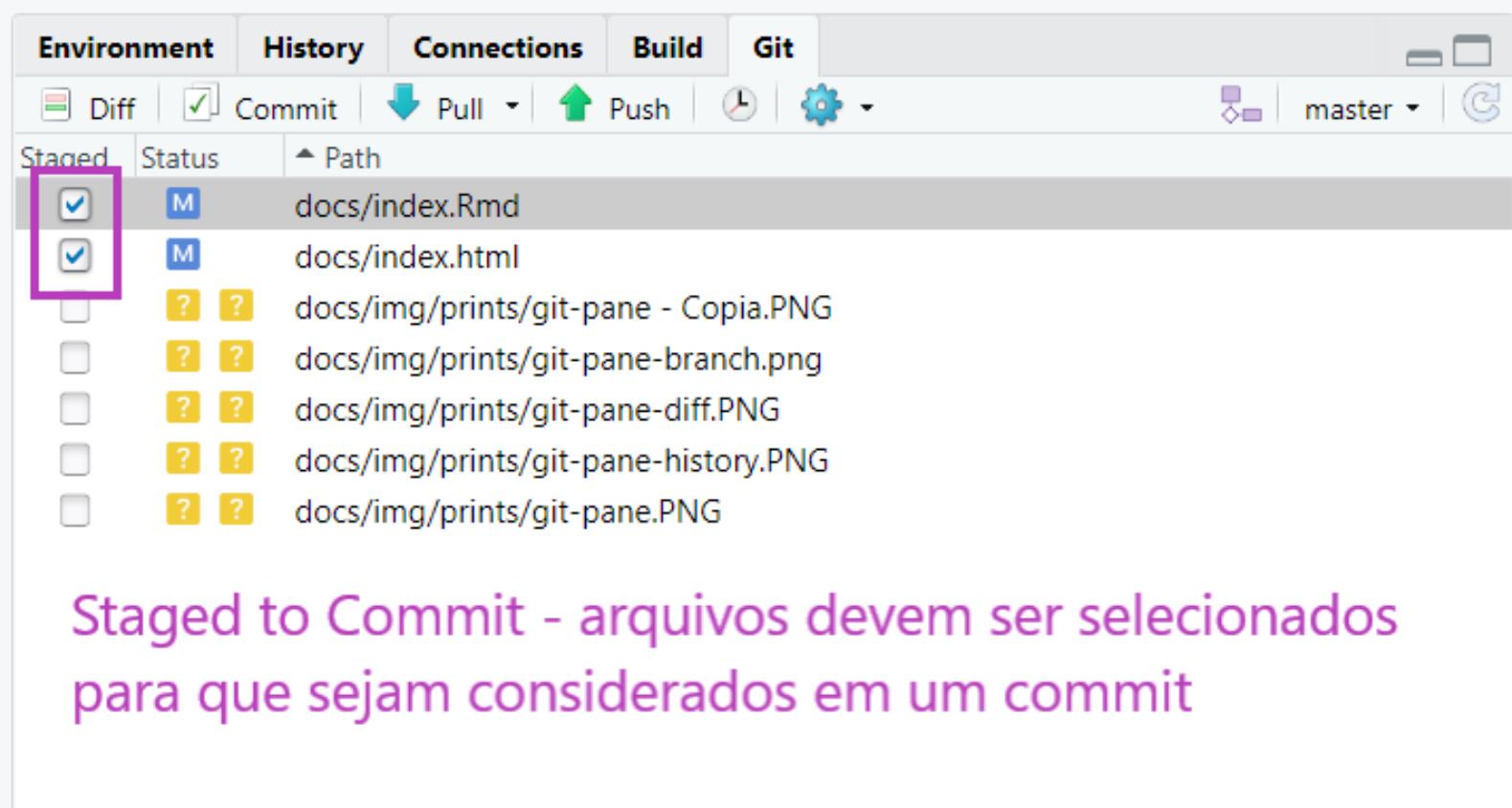
- Git Panel (top left):** Shows the "Review Changes" window with the "History" tab selected. It displays a list of commits from the "master" branch, with the most recent commit being the HEAD of the master branch. The commit message is "adiciona exemplos, muda títulos".
- History Tab (top center):** Shows a detailed history of changes, listing commits by Author, Date, SHA, and Subject.
- Diff View (bottom):** Shows the diff between the current state and the previous commit. The file being compared is "docs/index.Rmd". The diff highlights changes in lines 96-105, which involve adding sections for Git, GitHub, and working with RStudio and GitHub.

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	f5211570

```
@@ -96,10 +96,10 @@ class: split-33 with-border
96 96 ## Pré-requisitos
97 97 ]]
98 98 .row.bg-main2[.content[
99 99 ## Configurando o `Git`
100 100 ## Configurando o Git
101 101 .row.bg-main3[.content[
102 102 ## Configurando o `GitHub`
103 103 ## Configurando o GitHub
104 104 .row.bg-main4[.content[
105 105 ## Trabalhando com projetos no `Rstudio` + `GitHub`
```

1.13 Interface Gráfica RStudio

Git pane - Staged



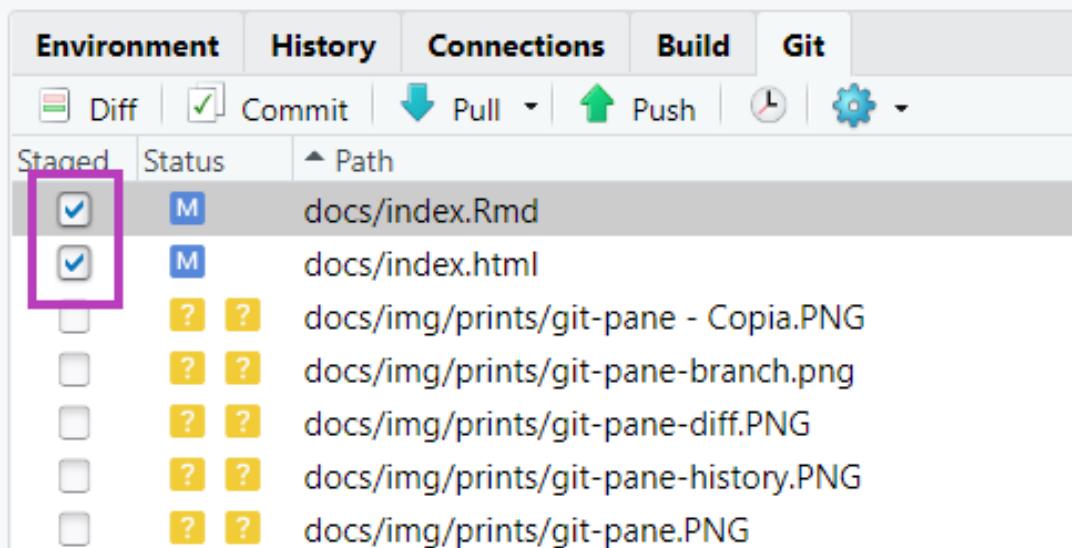
The screenshot shows the RStudio interface with the 'Git' tab selected in the top navigation bar. Below the navigation bar is a toolbar with icons for Diff, Commit, Pull, Push, and settings. To the right of the toolbar is a dropdown showing 'master'. The main area is the 'Git pane - Staged' view, which lists files with their status (M for modified) and path. The first two files, 'docs/index.Rmd' and 'docs/index.html', have their checkboxes checked, indicating they are staged for commit. A purple box highlights the checkboxes for these two files.

Staged	Status	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

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

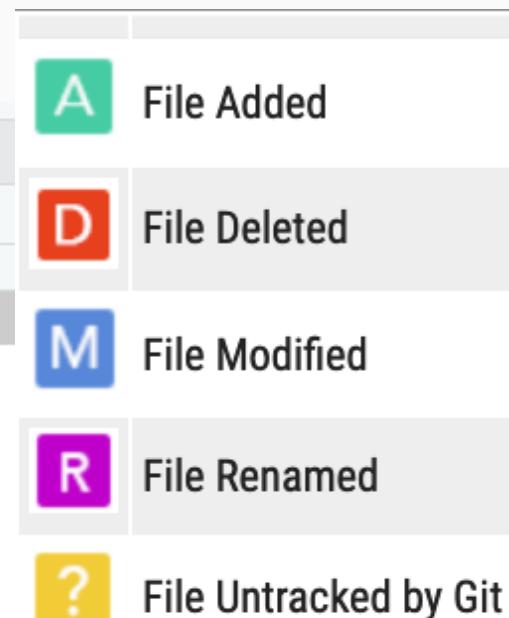
1.13 Interface Gráfica RStudio

Git pane - Staged e File status



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. The main area is titled 'Staged' and contains a list of files with their status indicators:

	Status	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



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

1.13 Interface Gráfica RStudio

Git pane - Commit

The screenshot shows the RStudio interface with the "Review Changes" tab selected. In the top left, there's a tree view of staged files under the "Changes" tab, showing paths like "docs/index.Rmd" and "docs/index.html". The bottom half shows the "Differences" tab with a diff view between two code snippets. The left snippet is from line 610 to 614, and the right snippet is from line 615 to 624. The diff highlights changes: new lines in green, deleted lines in red, and modified lines in yellow. A commit message box is open on the right, containing the text "adiciona slides sobre o git pane". Below the commit message box, there are buttons for "Amend previous commit" and "Commit".

Staged files

Commit message: adicione uma mensagem que seja útil para entender o que o commit faz

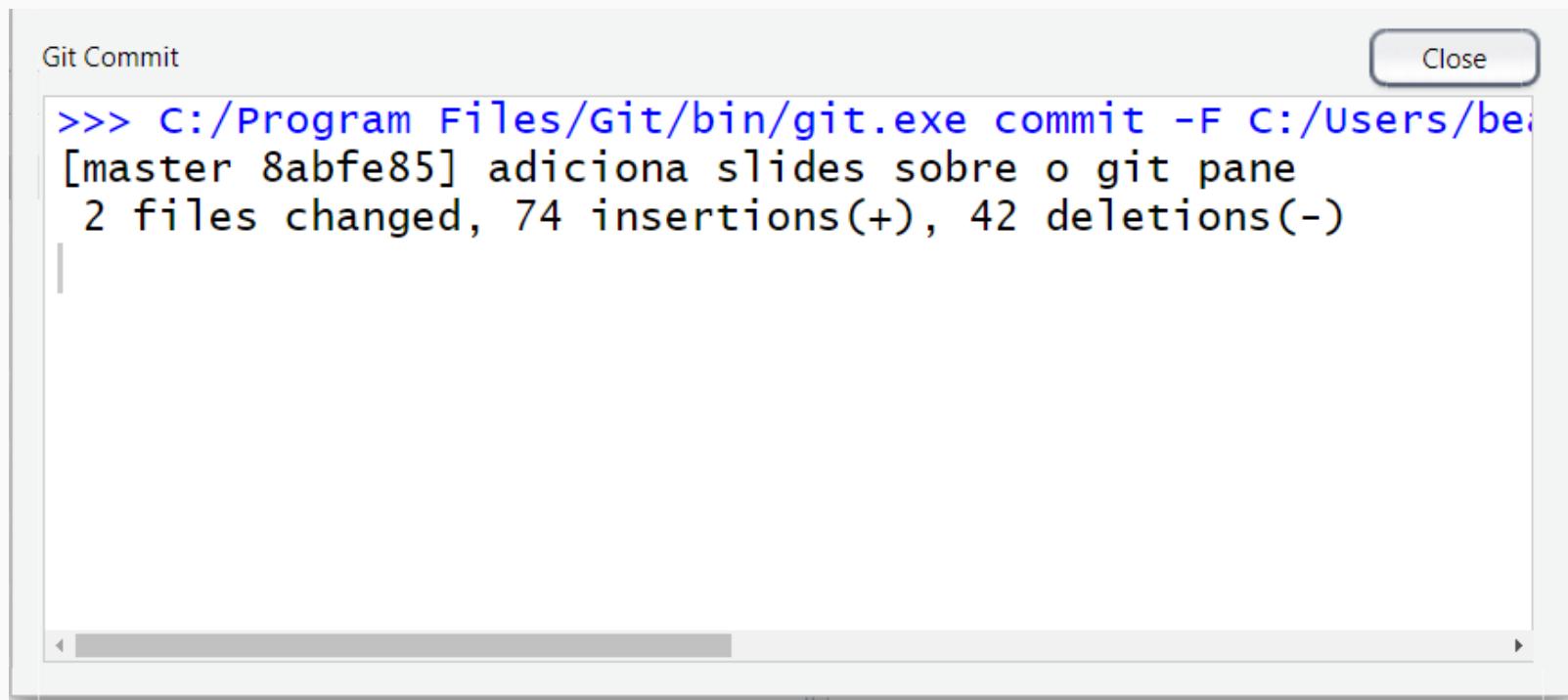
Mudanças verificadas: em verde, são novas linhas. em vermelho, são linhas retiradas/alteradas

```
@@ -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 ````
```

1.13 Interface Gráfica RStudio

Git pane - Commit



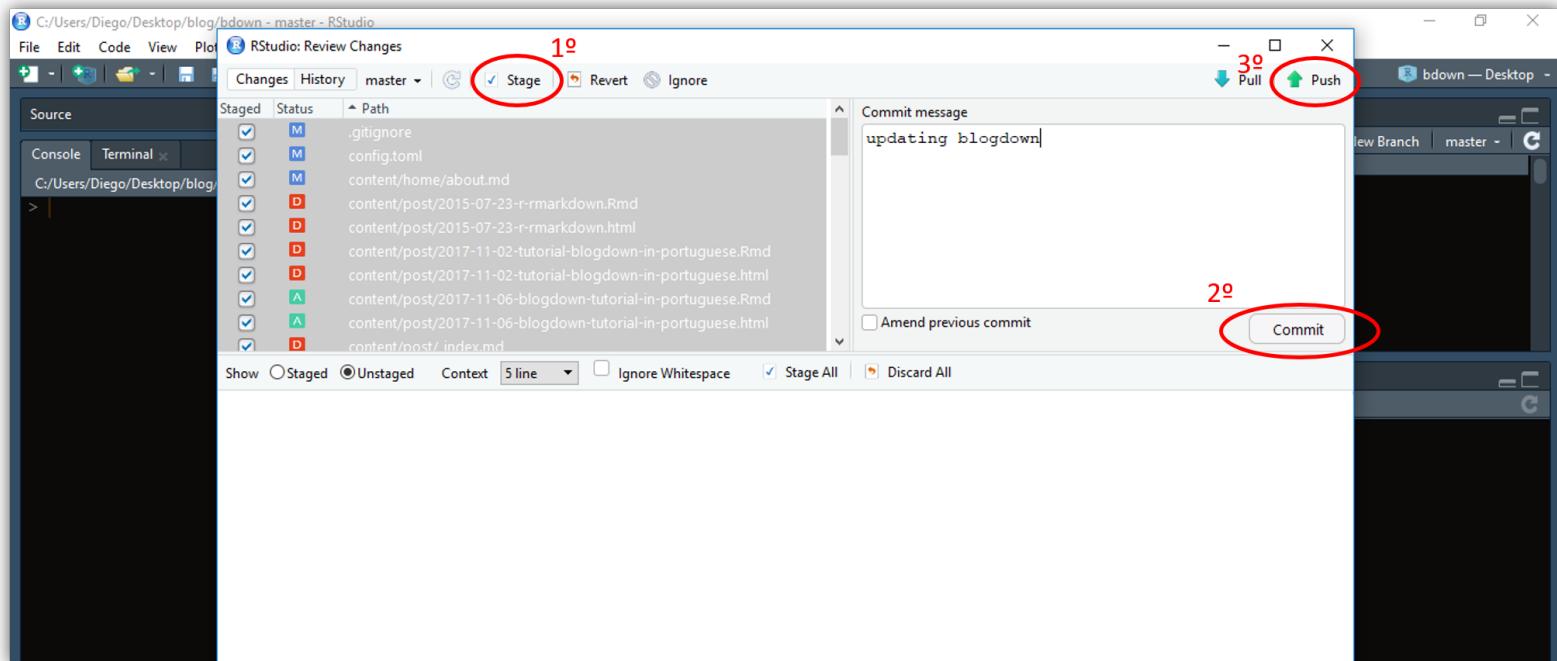
A screenshot of a 'Git Commit' dialog box from RStudio. The title bar says 'Git Commit' and there is a 'Close' button in the top right corner. The main area contains a command-line interface output:

```
>>> 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 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. GitHub for Windows https://windows.github.com GitHub for Mac https://mac.github.com Git distributions for Linux and POSIX systems are available on the official Git SCM web site. Git for All Platforms http://git-scm.com	MAKE CHANGES Review edits and craft a commit transaction \$ git status Lists all new or modified files to be committed \$ git diff Shows file differences not yet staged \$ git add [file] Schedules the file in preparation for versioning \$ git diff --staged Shows file differences between staging and the last file version \$ git reset [file] Unstages the file, but preserve its contents \$ git commit -m "[descriptive message]" Records file snapshots permanently in version history
CONFIGURE TOOLING Configure user information for all local repositories \$ git config --global user.name "[name]" Sets the name you want attached to your commit transactions \$ 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	GROUP CHANGES Name a series of commits and combine completed efforts \$ git branch Lists all local branches in the current repository \$ 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
CREATE REPOSITORIES Start a new repository or obtain one from an existing URL \$ git init [project-name] Creates a new local repository with the specified name \$ git clone [url] Downloads a project and its entire version history	

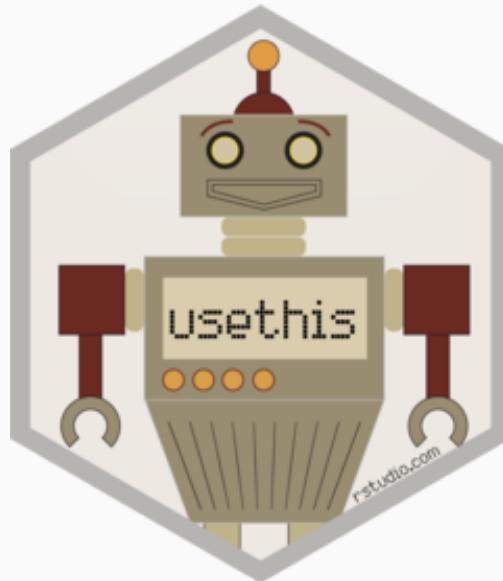
[*] https://github.github.com/training-kit/downloads/pt_BR/github-git-cheat-sheet.pdf

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.github.io



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