

# Introdução ao tidyverse

## 0 Controle de versão, Git e GitHub

xaringan [presentation ninja]

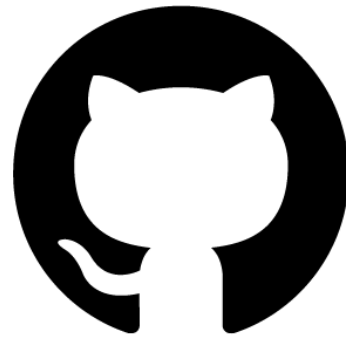
---

Maurício Vancine

25/04/2019



**git**



**GitHub**

# 0 Controle de versão, Git e GitHub

## Conteúdo

0.0 Verificar os notebooks

0.1 Controle de versão

0.2 Git

0.3 GitHub

0.4 Funcionamento do controle de versão

0.5 Fork

0.6 Clone

0.7 Add e Commit

0.8 Push

0.9 Pull

0.10 Controle de versão no RStudio



Vamos conferir os notebooks?!

# Software e Hardware



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

# 0.0 Verificar os notebooks

## Hardware

1 Notebook - Windows, MacOS ou Linux

2 Acesso à internet

# 0.0 Verificar os notebooks

Softwares

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



# 0.0 Verificar os notebooks

Softwares

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





# 0.0 Verificar os notebooks

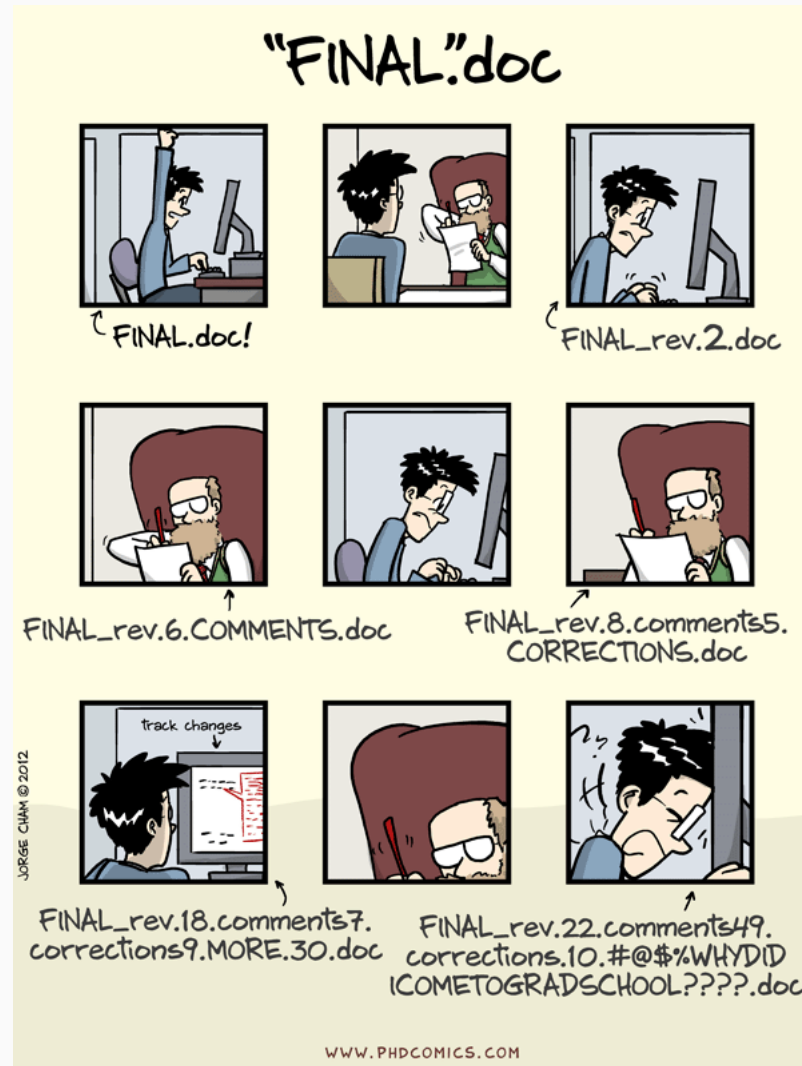
Softwares

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



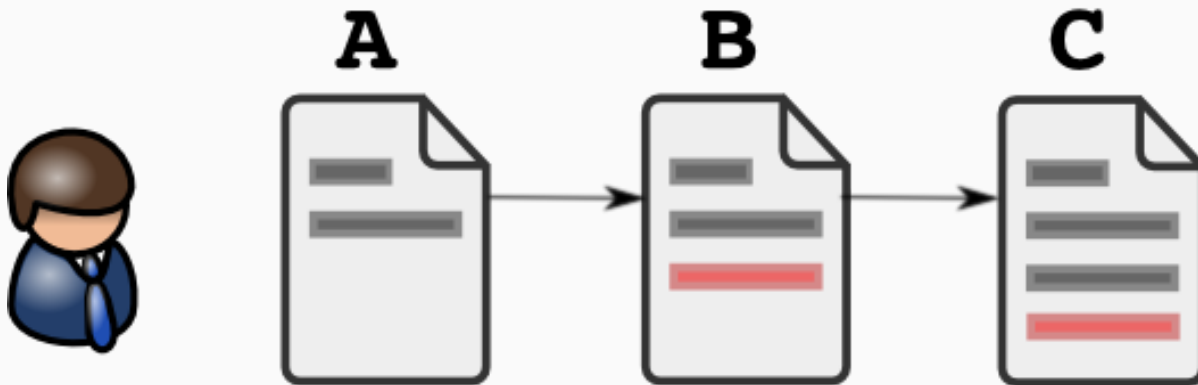
Tudo funcionando? Então bora!

# 0.1 Controle de versão



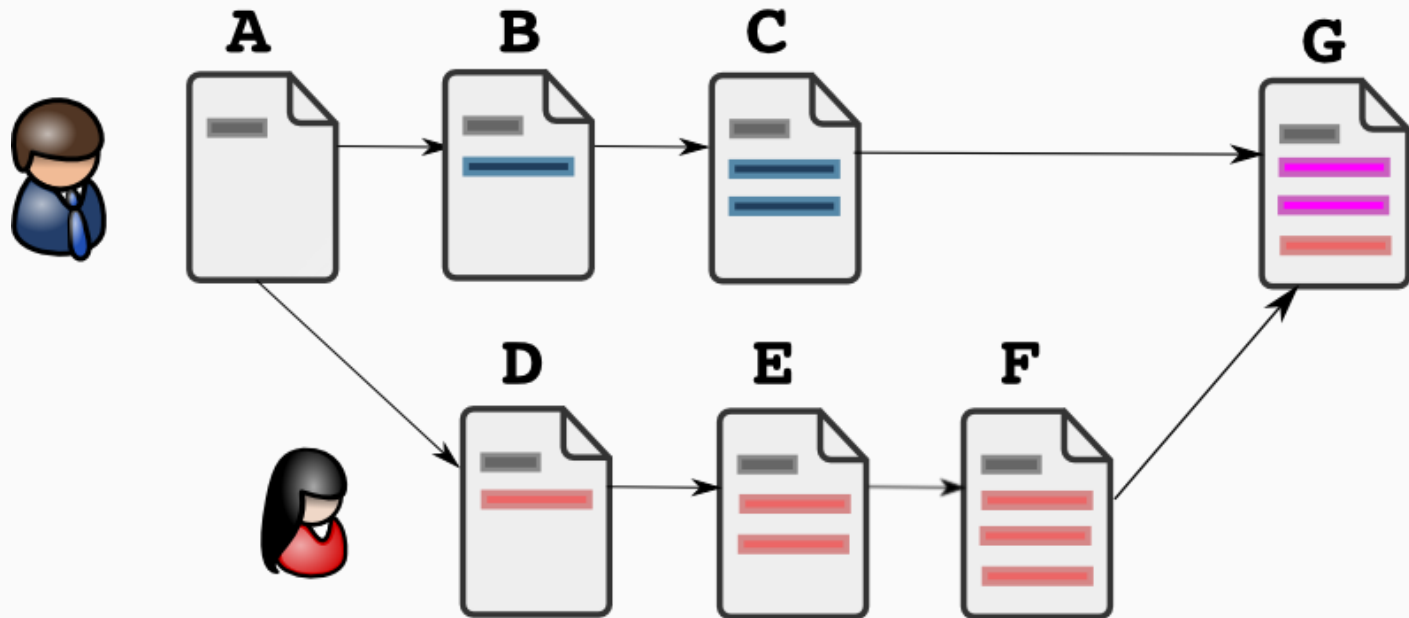
# 0.1 Controle de versão

Manejar projetos individualmente



# 0.1 Controle de versão

Manejar projetos compartilhados



## 0.2 Git

Software que faz o controle de versão

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



## 0.3 GitHub

### Repositório remoto

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



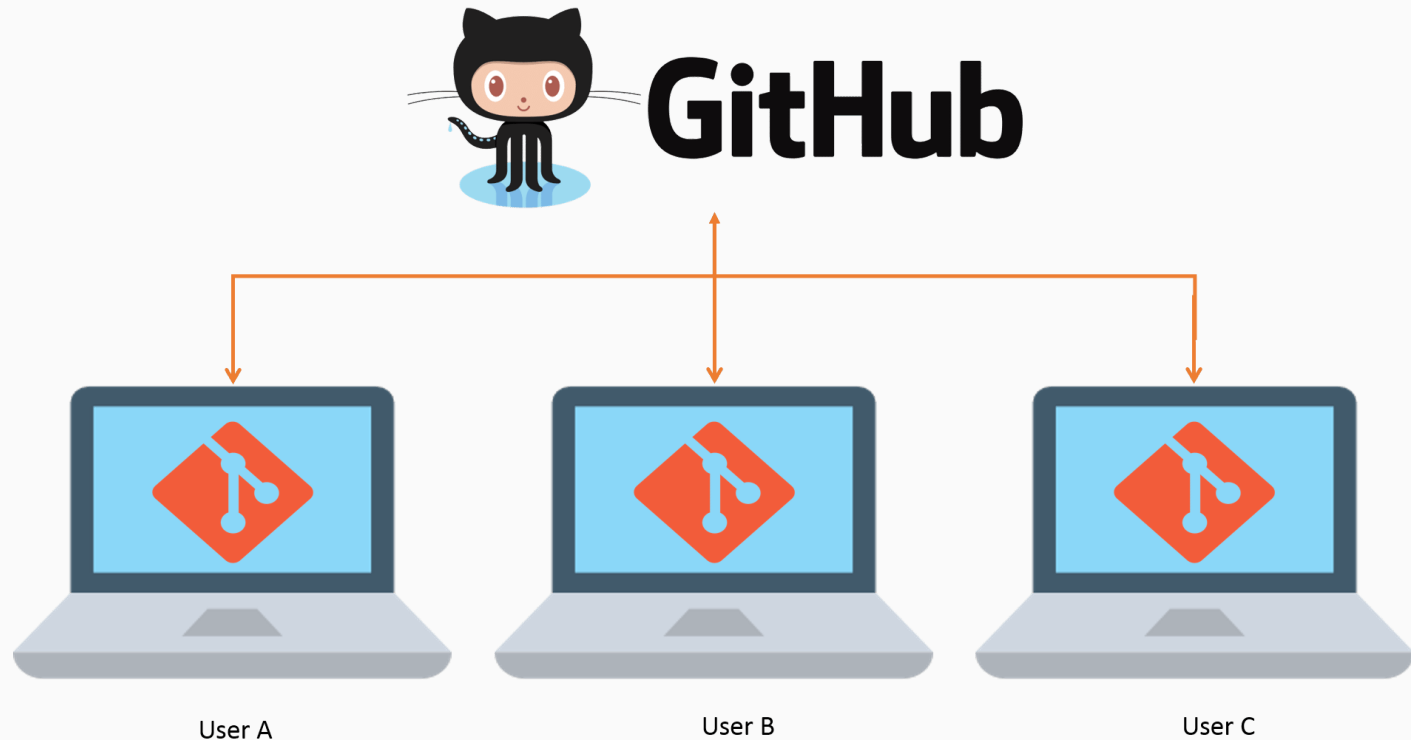
# Recapitulando



## 0.3 GitHub

**Git:** software que faz o controle de versão

**GitHub:** repositório remoto que hospeda os arquivos versionados



## 0.3 GitHub

Vamos criar uma conta no GitHub (caso não possuam)

<https://github.com>

## 0.3 GitHub

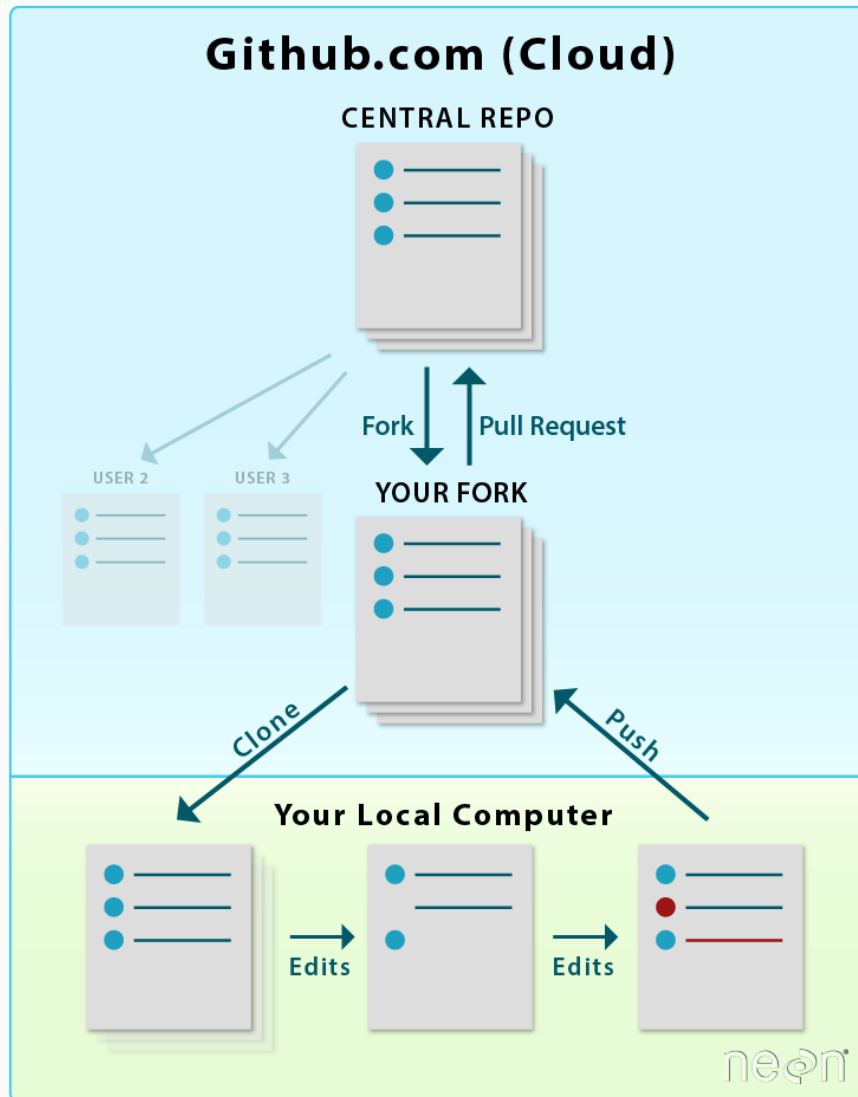
**Salvem essas informações!!!**

**username:** mauriciovancine

**email:** mauricio.vancine@gmail.com

**senha:** !@#\$%^&\*+ (e aqui ele é muito chato...)

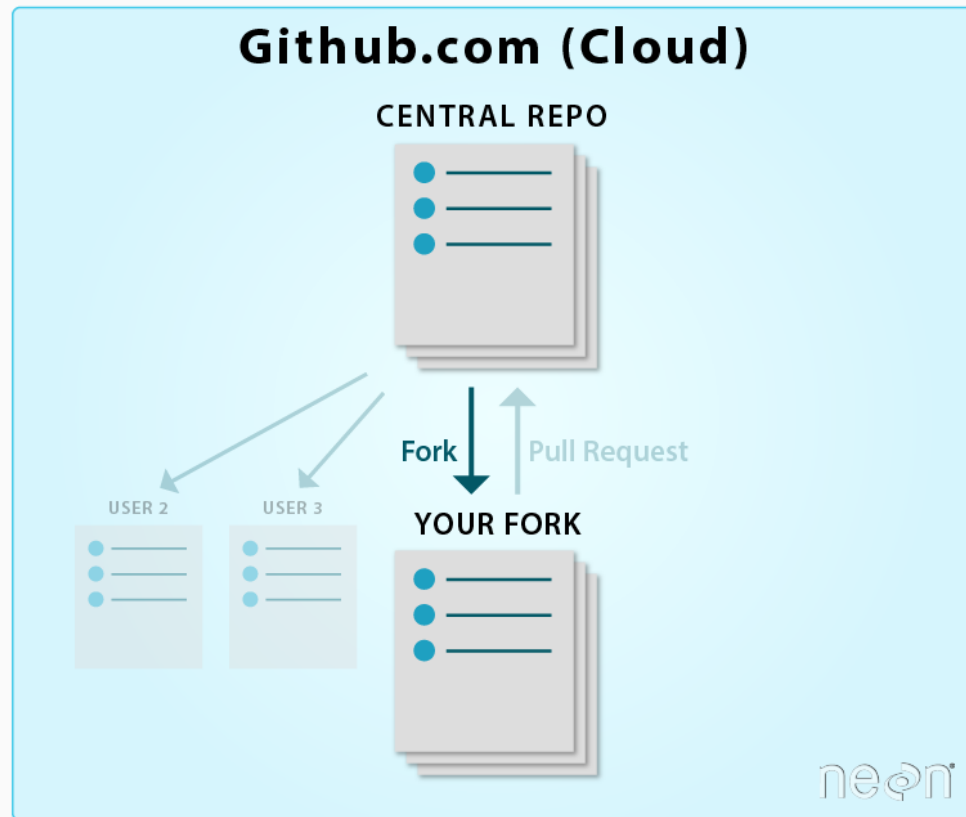
# 0.4 Funcionamento do controle de versão



Então vamos lá!

# 0.5 Fork

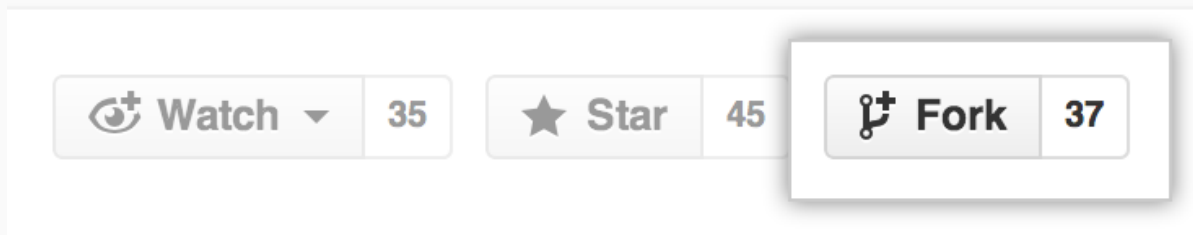
Fork: cópia de um repositório remoto para o seu GitHub



# 0.5 Fork

Forken o repositório do minicurso para o GitHub de vocês

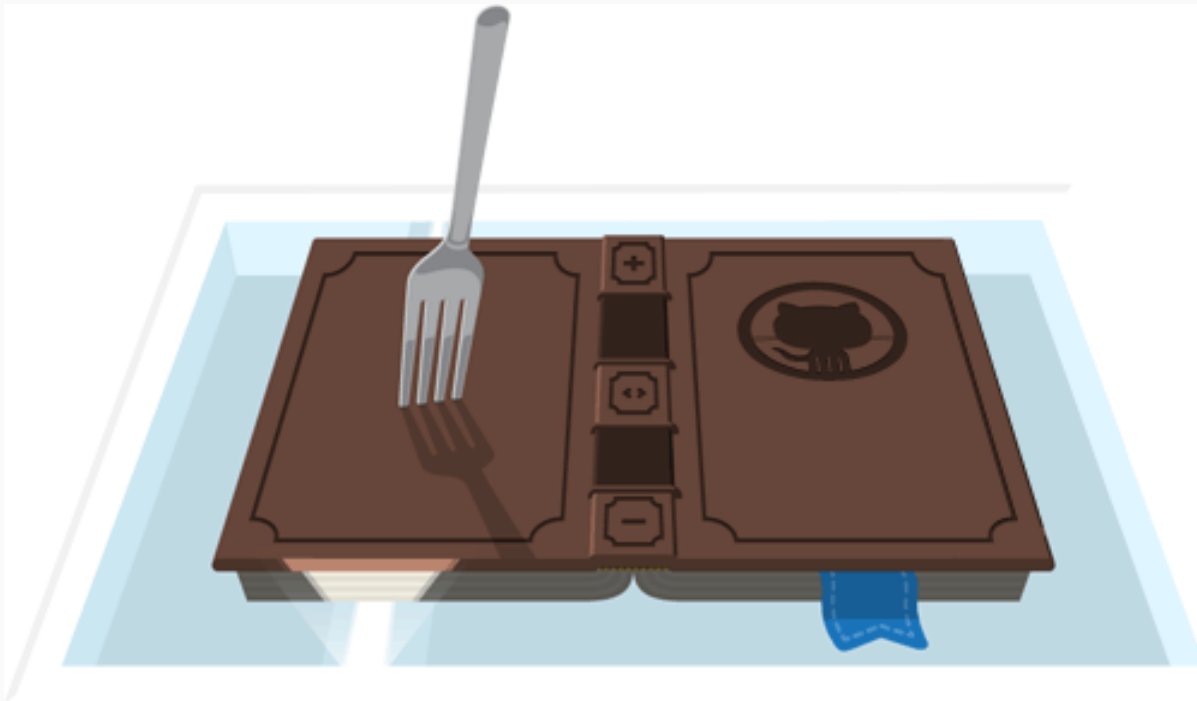
<https://github.com/mauriciovancine/minicurso-tidyverse>



# 0.5 Fork

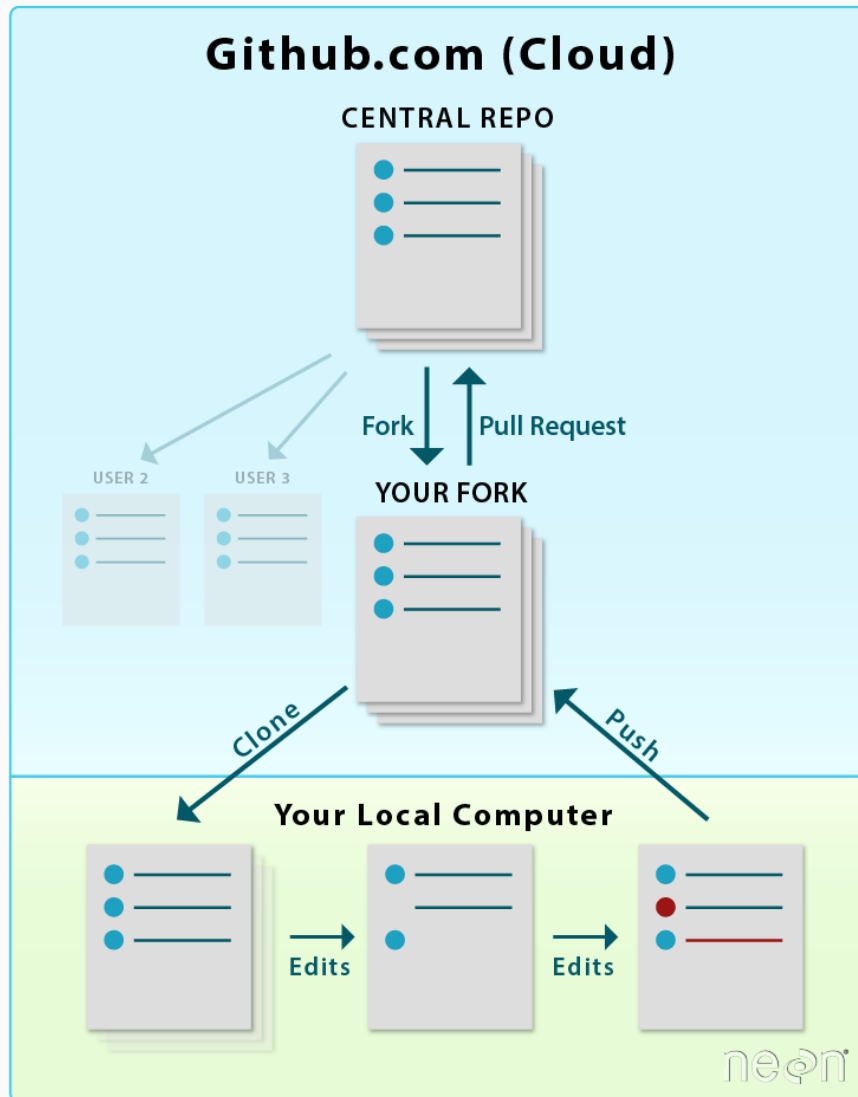
Forken o repositório do minicurso para o GitHub de vocês

<https://github.com/mauriciovancine/minicurso-tidyverse>



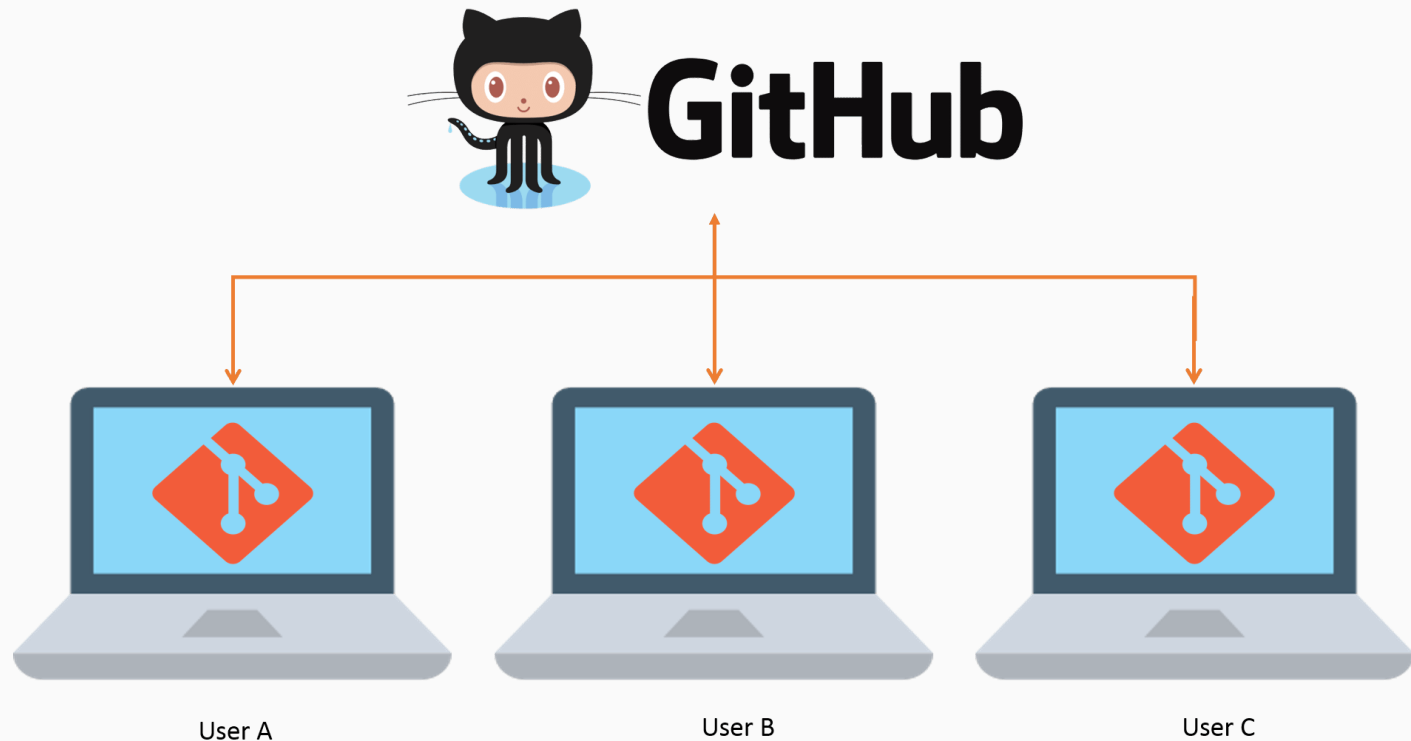


# 0.4 Funcionamento do controle de versão



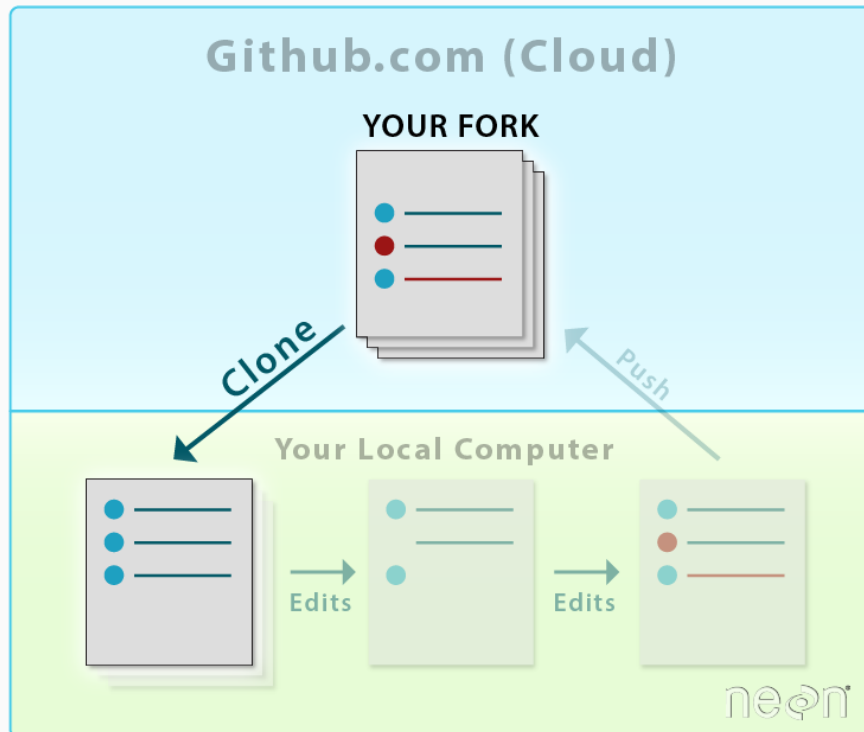
# Repositório local

Agora vamos criar uma pasta chamada `github` onde o repositório da disciplina irá ficar armazenado



# 0.6 Clone

Fazer o download do repositório remoto para o repositório local



Agora precisamos trazer esses arquivos para  
nosso notebook

E como faremos isso?

# 0.6 Clone

## Diretamente do repositório remoto

The screenshot shows the GitHub interface for the repository 'mjoness01 / DI16-NEON-participants'. The repository is forked from 'NEON-WorkWithData/DI16-NEON-participants'. The 'Clone or download' button is highlighted with a purple circle. The repository has 4 commits, 1 branch, 0 releases, and 1 contributor. The current branch is 'gh-pages'. The repository description is 'A repository for all participant code & documents during the 2016 NEON Data Institute'. The file list shows 'participants' (file for week 3 activity w/ example file --to be updated, 4 minutes ago), 'README.md' (Initial commit, 2 days ago), and 'README.md' (file for week 3 activity w/ example file --to be updated, 4 minutes ago).

GitHub repository page for **mjoness01 / DI16-NEON-participants**, forked from **NEON-WorkWithData/DI16-NEON-participants**.

Navigation: [Pull requests](#) [Issues](#) [Gist](#)

Actions: [Unwatch](#) 1 [Unstar](#) 1 [Fork](#) 1

Repository tabs: [Code](#) [Pull requests](#) 0 [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)

A repository for all participant code & documents during the 2016 NEON Data Institute — [Edit](#)

Repository stats: 4 commits, 1 branch, 0 releases, 1 contributor

Branch: **gh-pages** [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

This branch is 2 commits behind **NEON-WorkWithData:gh-pages**. [Pull request](#) [Compare](#)

**mjoness01** file for week 3 activity w/ example file --to be updated Latest commit 56f44c3 4 minutes ago

File	Commit	Time
<a href="#">participants</a>	file for week 3 activity w/ example file --to be updated	4 minutes ago
<a href="#">README.md</a>	Initial commit	2 days ago

[README.md](#)

# 0.6 Clone

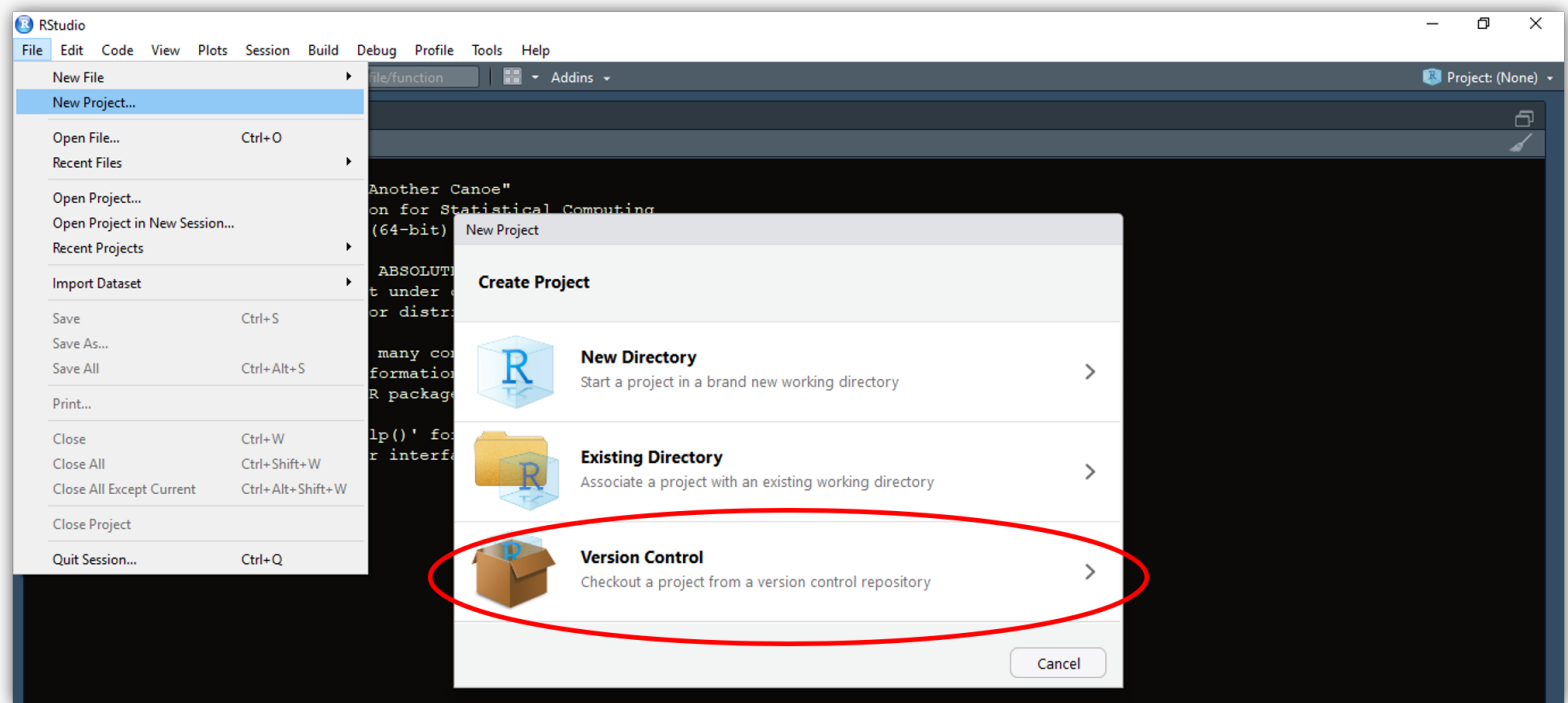
## Git e RStudio



# Abram o RStudio

# 0.6 Clone

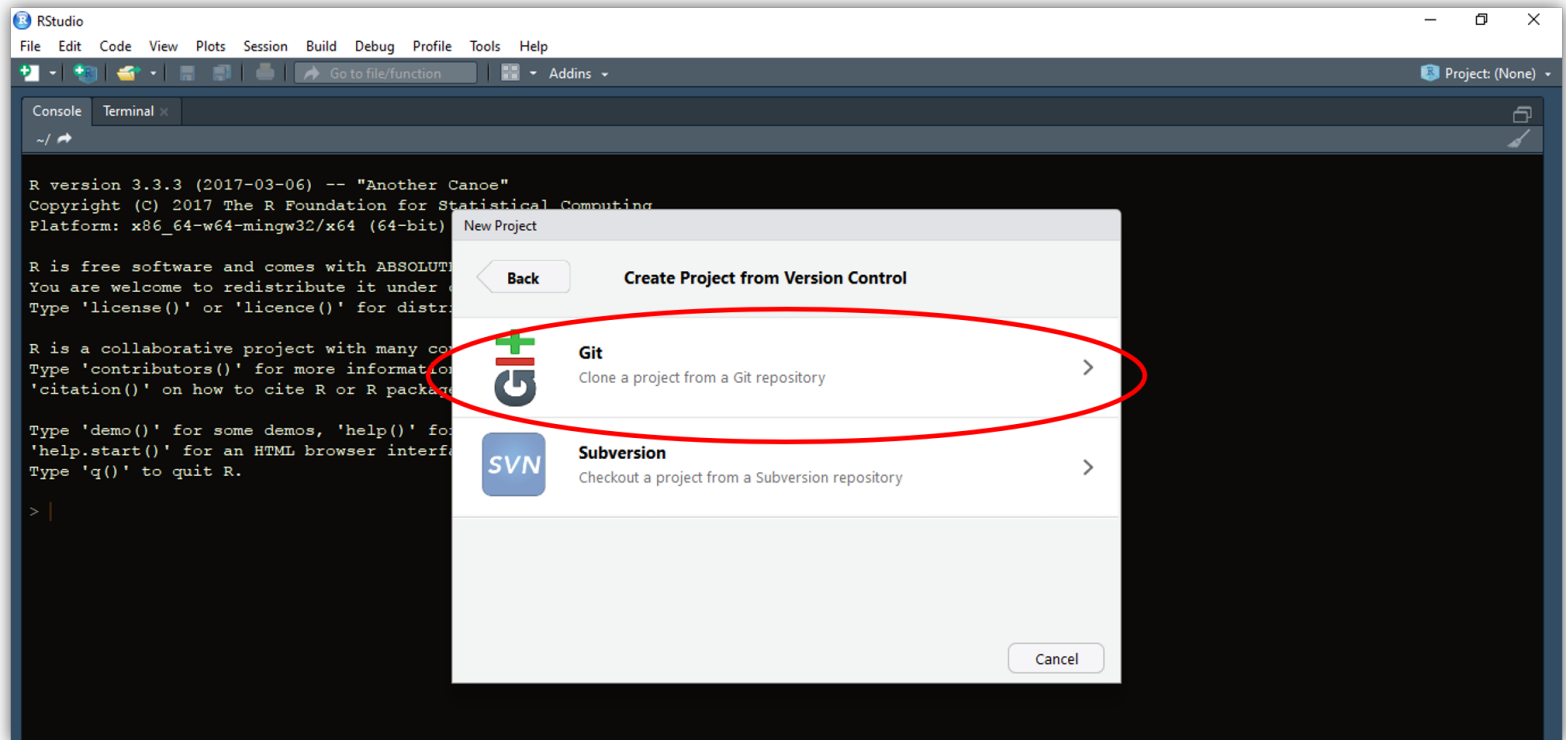
## Criar um projeto R com controle de versão





# 0.6 Clone

## Escolher clonar repositório do GitHub



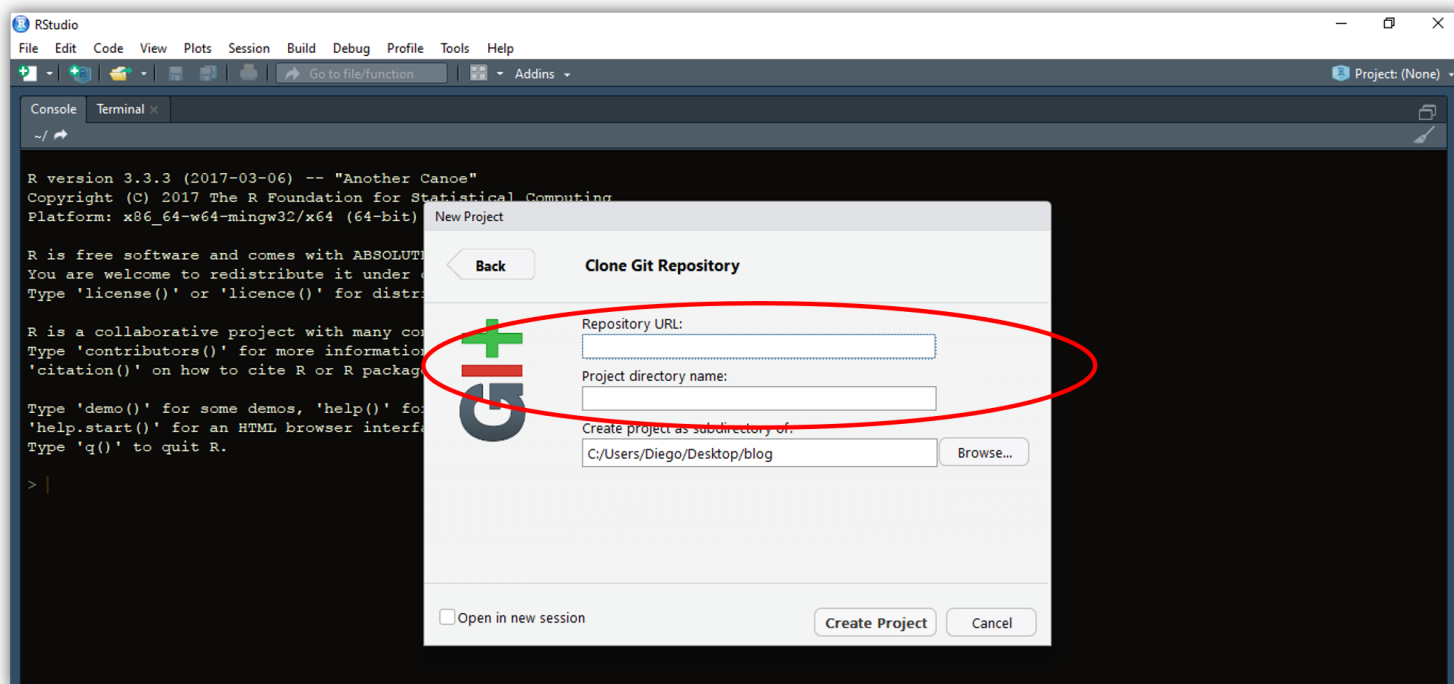
# 0.6 Clone

Endereço do repositório remoto do GitHub de vocês

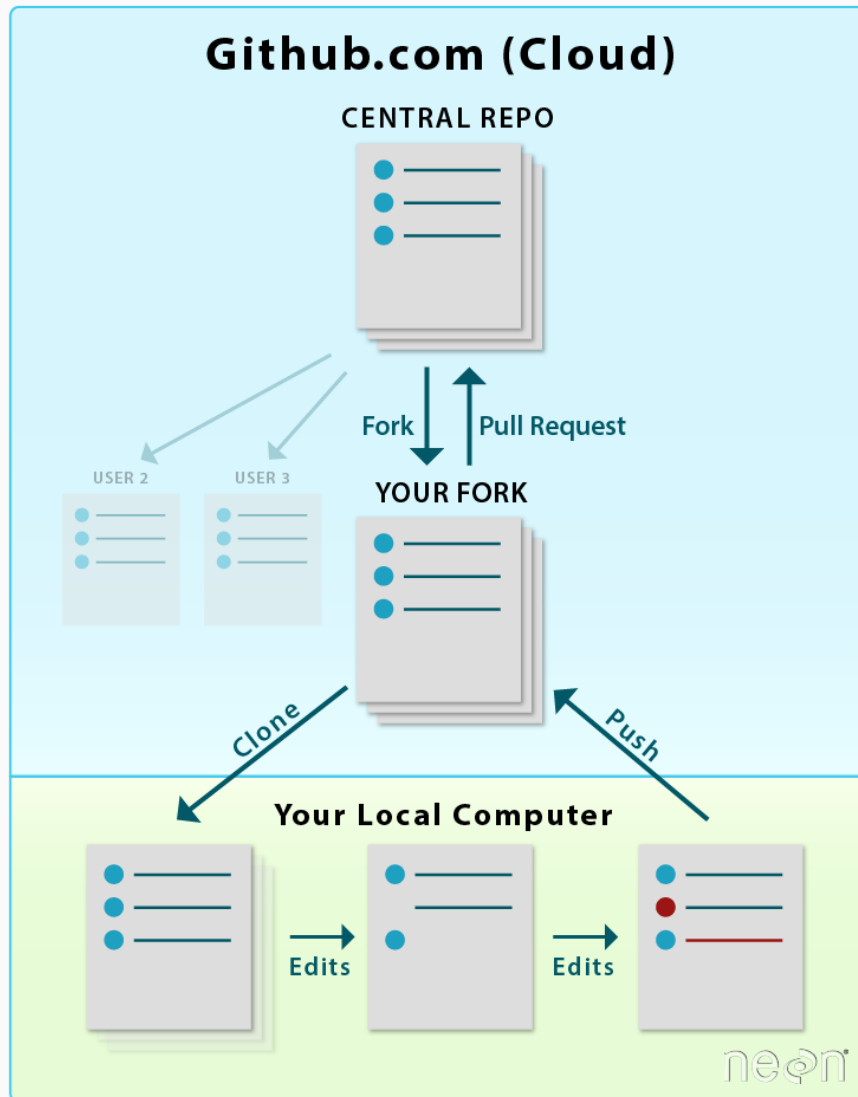
<https://github.com/username/minicurso-tidyverse>

Diretório da pasta `github`

`/home/mude/data/github`



# 0.6 Clone



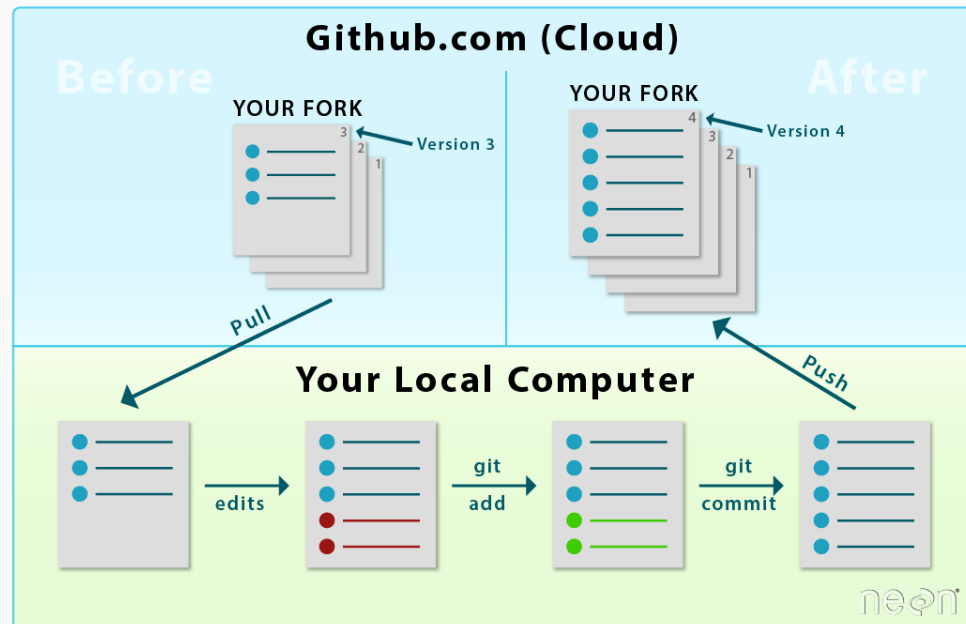
# 0.7 Add e Commit

## Add

**Adicionar** as mudanças após edições (stage)

## Commit

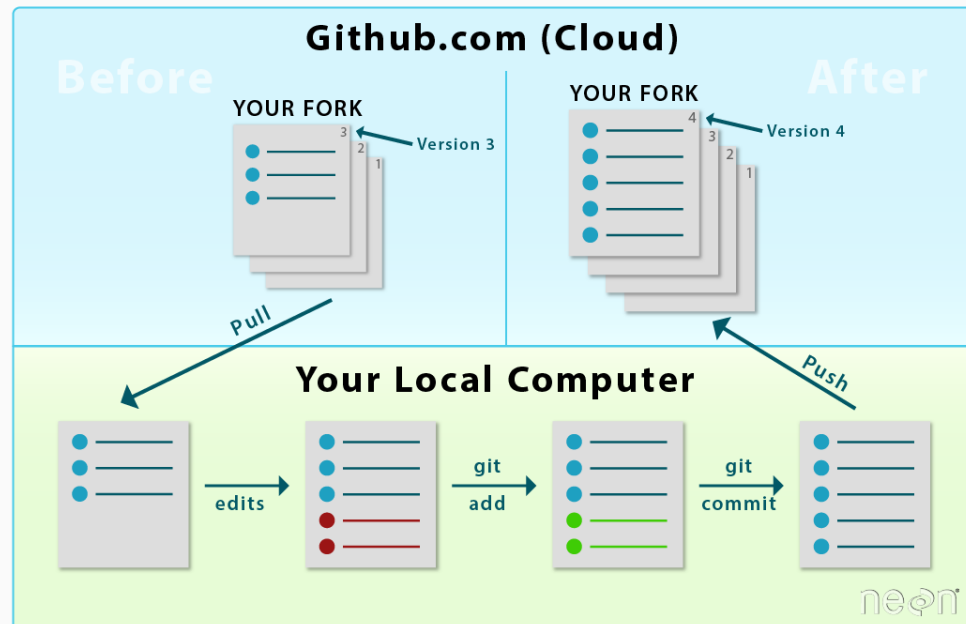
**Criar** uma nova versão no repositório local



# 0.8 Push

## Push

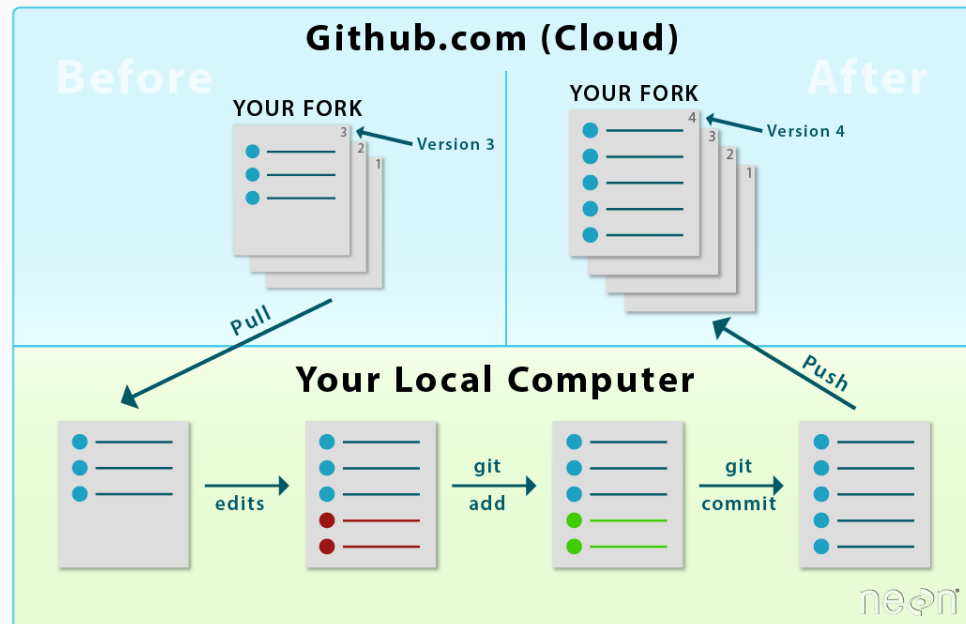
Enviar uma nova versão do **repositório local** para o **repositório remoto (GitHub)**



# 0.9 Pull

## Pull

Enviar uma nova versão do **repositório remoto (GitHub)** para o **repositório local**

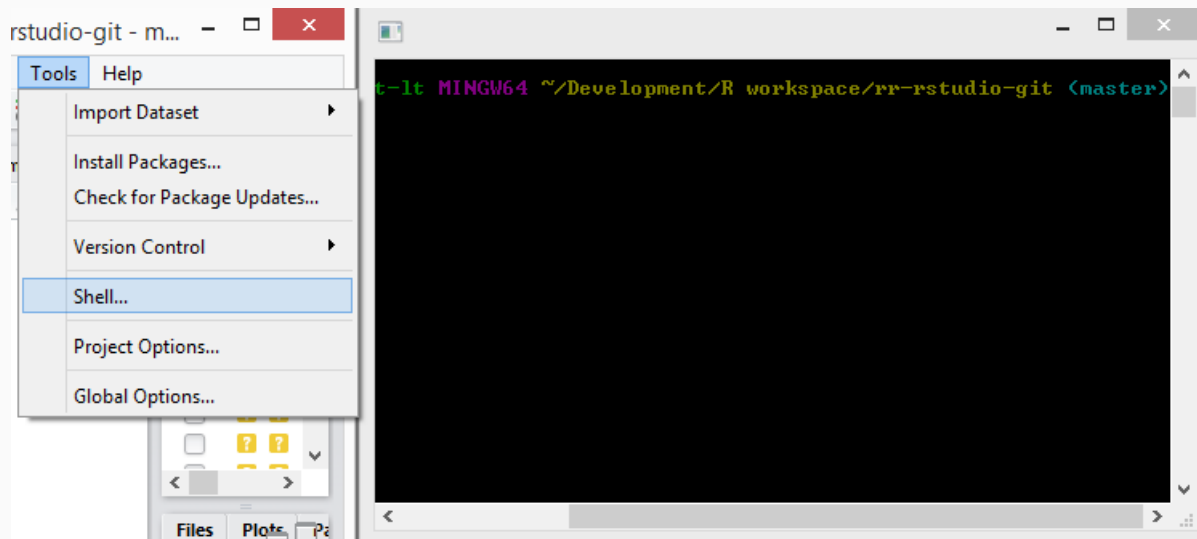


# 0.10 Controle de versão no RStudio

## RStudio

Configurar o nome de usuário, e-mail e senha

```
## add user and email  
git config --global user.name "seu_usuario"  
git config --global user.email "seu_email"
```

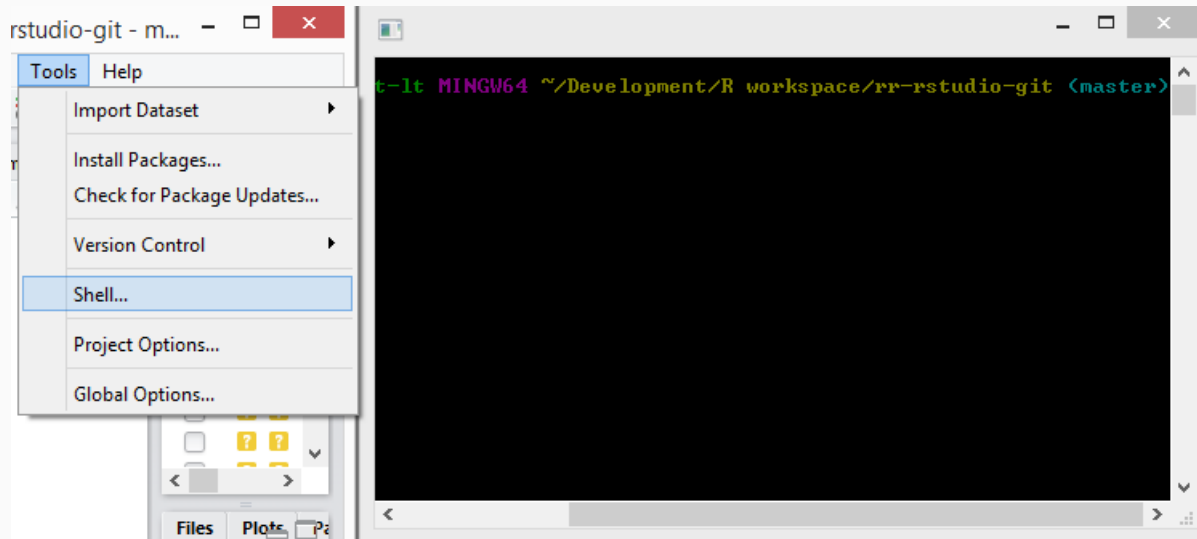


# 0.10 Controle de versão no RStudio

## RStudio

Configurar o nome de usuário, e-mail e senha

```
# ssh key  
eval $(ssh-agent -s)  
ssh-add ~/.ssh/id_rsa # digite sua senha
```

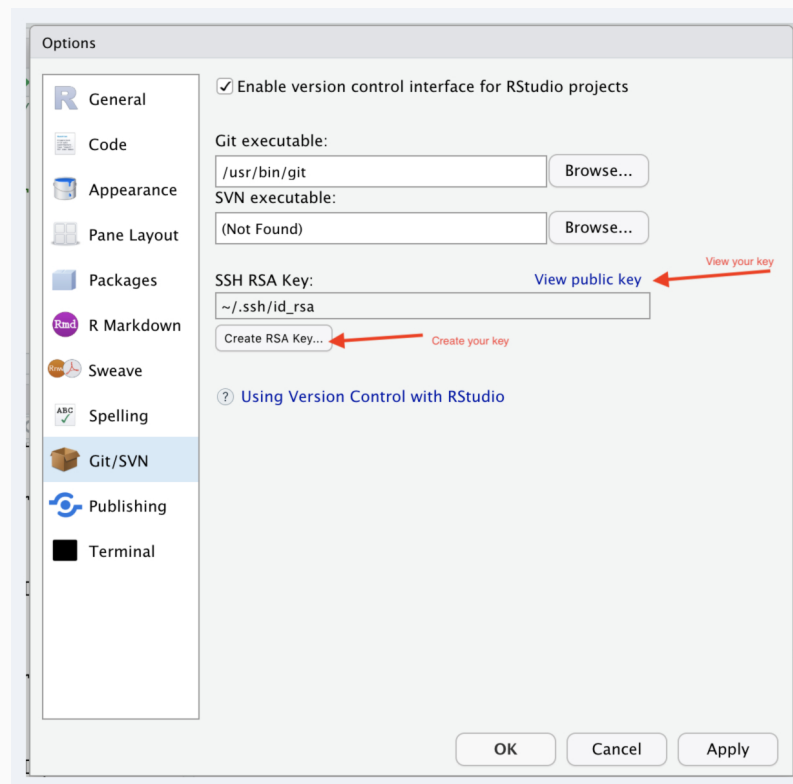




# 0.10 Controle de versão no RStudio

## RStudio

### Criar o SSH



# 0.10 Controle de versão no RStudio

## RStudio

### Adicionar o SSH ao GitHub

The screenshot shows the GitHub 'Personal settings' page. On the left sidebar, 'SSH keys' is highlighted with a red arrow and the number 1. The main content area has a header 'Need help? Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#)'. Below this is a section titled 'SSH keys' with a button 'Add SSH key' indicated by a red arrow and the number 2. The text 'There are no SSH keys with access to your account.' is displayed. Below this is a section 'Add an SSH key' with a red arrow and the number 3 pointing to the 'Title' field, which contains 'mac-key'. The 'Key' field is a large text area containing a long SSH public key, with a red arrow and the number 4 pointing to it and the text '4. Paste ssh public key here'. At the bottom of the 'Key' field is a green 'Add key' button, indicated by a red arrow and the number 5.

Personal settings

- Profile
- Account settings
- Emails
- Notification center
- Billing
- SSH keys**
- Security
- Applications
- Personal access tokens
- Repositories
- Organizations

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

SSH keys

Add SSH key

There are no SSH keys with access to your account.

Add an SSH key

3. Enter any title here

Title

mac-key

4. Paste ssh public key here

Key

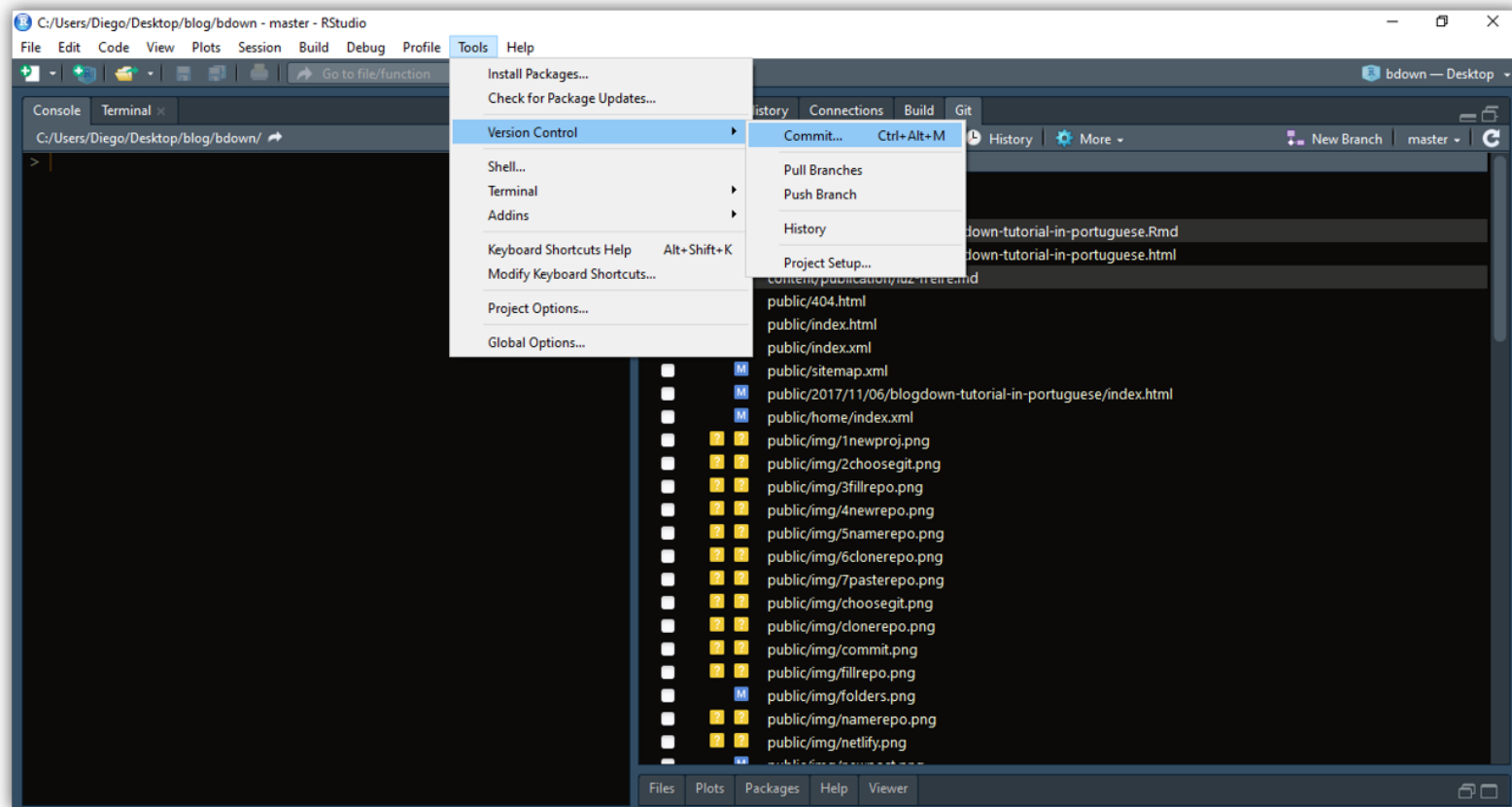
```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAjC29d2T7SaIPMASSyKdX5ZIV3vyxWky6l8ALjtklQr8nFT87ZaV6qyXz2QS7JZ2E7kPdJGwYZsaztjs4D7ZO1YmkPmEdunWXMDLKdxUTz1PHSmnVn/ew1/Ob0snxgZ/RziGq4fzscsRuKXSw/rWtEoFa0Fq392ZonVzeZllc1mpZhPYATvNNILm8eZ78PWe2tGyMx8GpZYdyj1lJn3c7LB7J6cl6z8K21p6eLR9R9QS+DHB7uOvVripFlp/MOes4meJToEpl4Jortg+0IKvLIRbszhbpuh2f2+M6Ly5uVE7rKlmQYHzt3lPIGBwmVxHbGRofxVhU/RLbWKvXvOSGtubGpYat@gmail.com
```

5

Add key

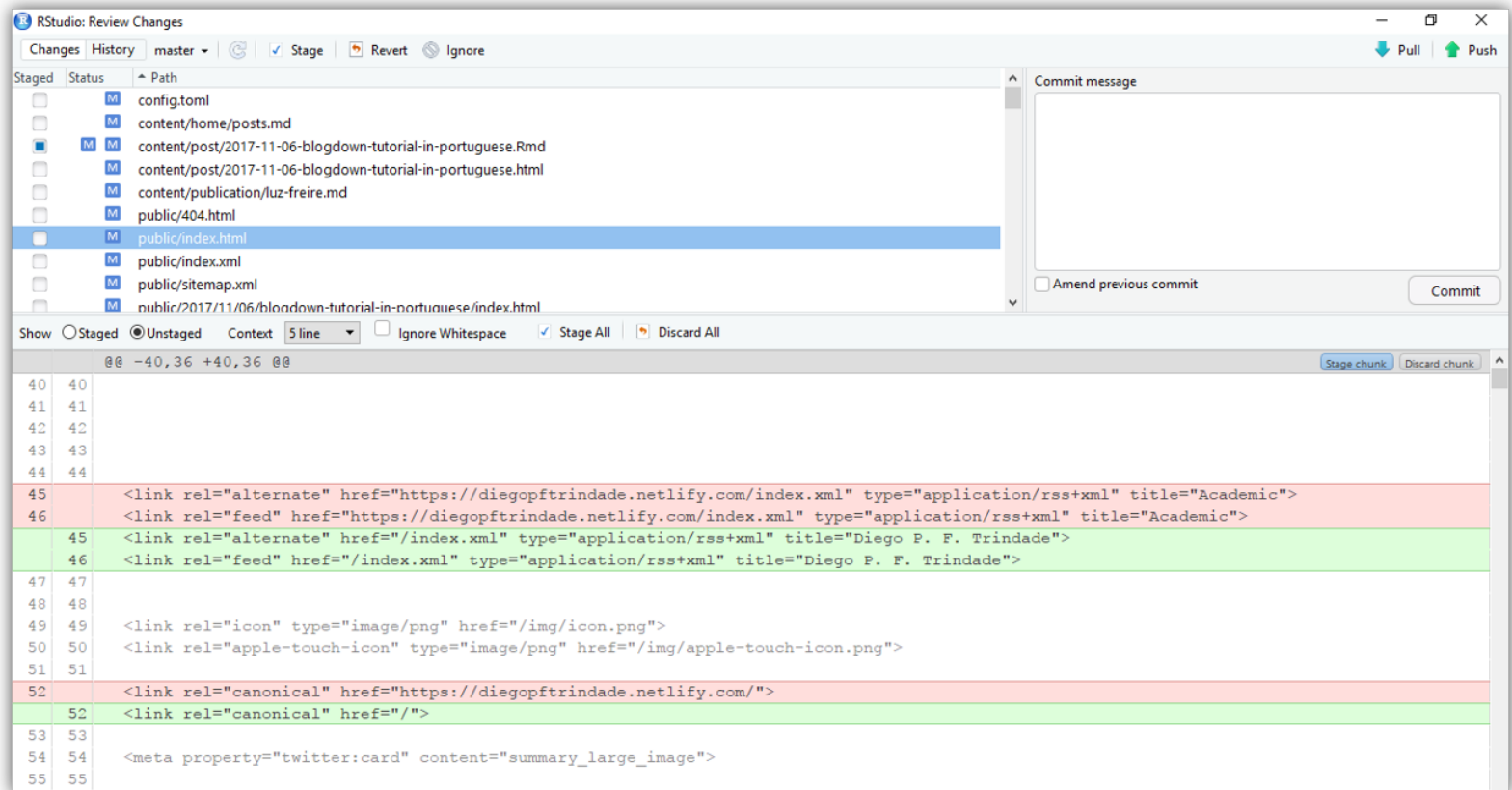
# 0.10 Controle de versão no RStudio

## RStudio - Abrir o commit



# 0.10 Controle de versão no RStudio

## RStudio - Verificar mudanças



# 0.10 Controle de versão no RStudio

## RStudio - Verificar mudanças

### Version Control with Git or SVN



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

Stage files:      Show file diff      Commit staged files      Push/Pull to remote      View History

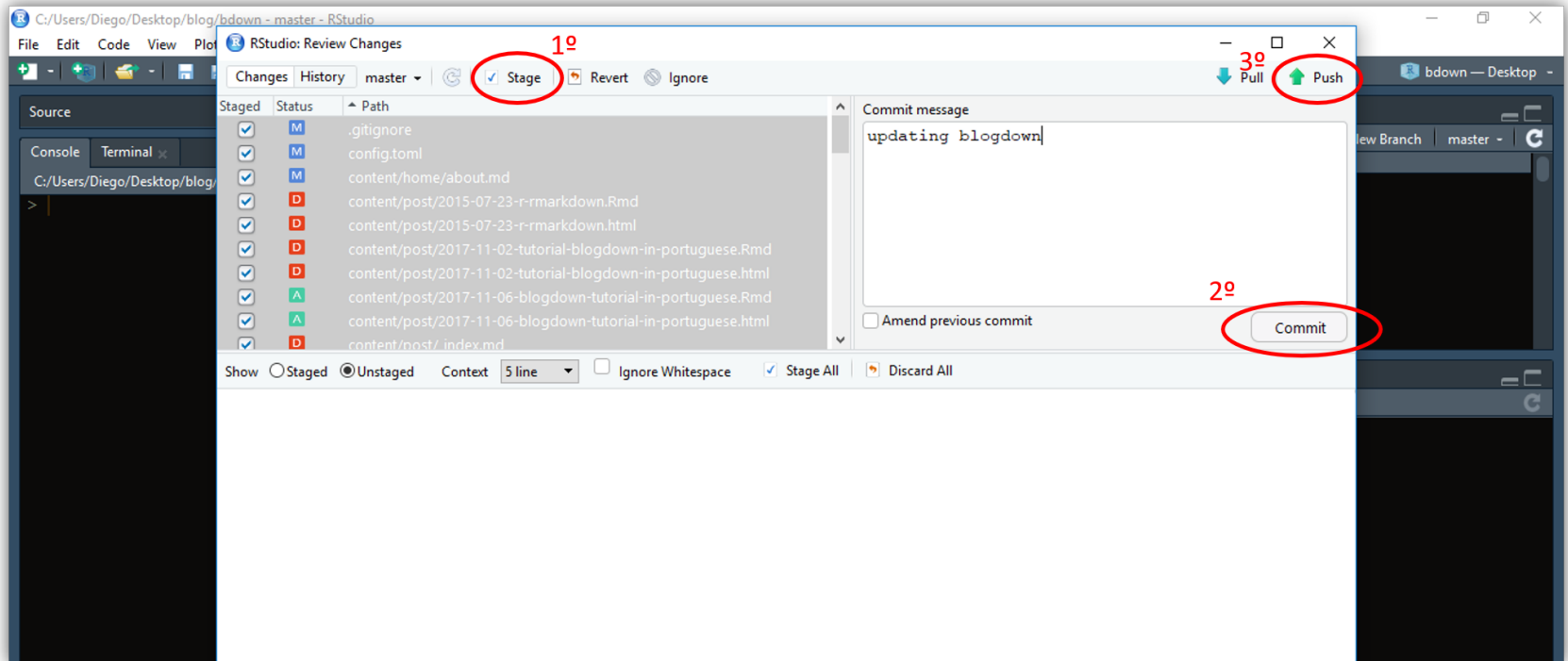
**A** Added  
**D** Deleted  
**M** Modified  
**R** Renamed  
**?** Untracked

Open shell to type commands

current branch

# 0.10 Controle de versão no RStudio

## RStudio - Stage, Commit e Push



# 0.10 Controle de versão no RStudio

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

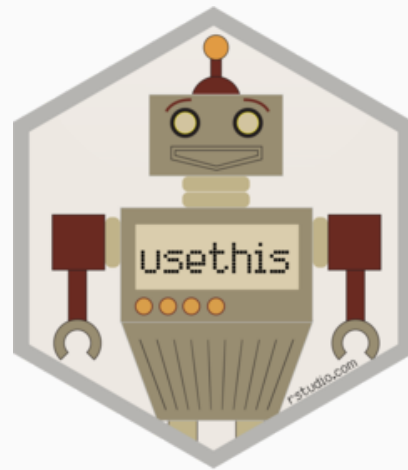
[\*] [https://github.github.com/training-kit/downloads/pt\\_BR/github-git-cheat-sheet.pdf](https://github.github.com/training-kit/downloads/pt_BR/github-git-cheat-sheet.pdf)

# 0.10 Controle de versão no RStudio

## Pacote [usethis](#)

Comandos diretos no R para versionar um repositório

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



## Material

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

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



Dúvidas?

# Maurício Vancine

Contatos:

✉ [mauricio.vancine@gmail.com](mailto:mauricio.vancine@gmail.com)

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

🔗 [mauriciovancine.netlify.com](https://mauriciovancine.netlify.com)

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