



Utilizando o Git e GitHub no RStudio

Beatriz Milz [✉](mailto:beatriz.milz@gmail.com) [@beatriz_milz](https://twitter.com/beatriz_milz) [GitHub](https://github.com/beatriz-milz)

Meetup R-Ladies GYN

2º semestre/2020

Atualizado em 28 de agosto de 2020.

Para ativar o modo tela cheia, pressione **ALT + F**, e depois **F**



Beatrix Milz

Contexto Acadêmico

- Doutoranda no Programa de Pós-Graduação em Ciência Ambiental (PROCAM)
 - Instituto de Energia e Ambiente - Universidade de São Paulo
- Equipe da Secretaria Executiva Editorial - Revista Ambiente & Sociedade
- Anteriormente:
 - Mestre em Ciências - UNIFESP;
 - Bacharel em Gestão Ambiental - EACH/USP

Comunidades de R



- Co-autora do pacote {dados}
- Co-organizadora: R-Ladies São Paulo ❤️
- Comitê organizador:
 - satRday São Paulo
 - LatinR
 - useR! 2021
- Instrutora The Carpentries



Sobre este material

- **Público-alvo:** pessoas que utilizam o R para análise de dados, tem um conhecimento básico sobre Git e GitHub e querem utilizar essas ferramentas diretamente do RStudio.
- Partindo do ponto que:
 - Você já conhece os conceitos básicos de Git
 - Você já sabe porque é interessante usar Git
- Paralelamente irei falar sobre como usar Git:
 - No RStudio Cloud
 - No RStudio instalado no seu computador
- A maior parte deste conteúdo é derivado [deste material](#), oferecido por mim em um meetup da R-Ladies São Paulo em Setembro/2019.



Comentários importantes

- Você pode usar o Git através:
 - Linha de comando :
 - git
 - terminal do RStudio
 - Interfaces gráficas:
 - GitHub Desktop
 - Sourcetree
 - Git Pane do RStudio
 - Entre outros!

Aqui apresento com o RStudio, mas escolha o que é mais confortável para você :)

- Independente de qual ferramenta você escolher, é importante qual comando do Git equivale a um "botão" na interface gráfica. É muito importante entender as equivalências!

Pré-requisitos



Para usar no Computador



R

- Download neste link
- Use a versão mais recente
😊





[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQs](#)
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\)OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

[Source code for all platforms](#)

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2019-07-05, Action of the Toes) [R-3.6.1.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to you to minimize network load.



RStudio

- Download neste link
- Use a versão mais recente
😊





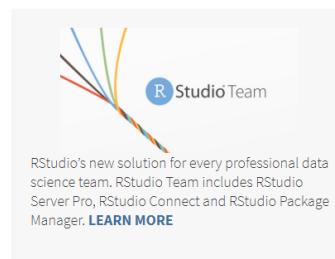
rstudio:conf 2020 Early Bird discount ends August 31! [Register Now](#)



Products Resources Pricing About Us Blogs

Choose Your Version of RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace. [Learn More](#) about RStudio features.



<p>RStudio Desktop Open Source License FREE DOWNLOAD</p>	<p>RStudio Desktop Commercial License \$995 per year BUY</p>	<p>RStudio Server Open Source License FREE DOWNLOAD</p>	<p>RStudio Server Pro Commercial License \$4,975 per year (5 Named Users) BUY</p>
<p>Learn More</p>	<p>Learn More</p>	<p>Learn More</p>	<p>Evaluation Learn More</p>



Git

- Download neste link





 **git** --local-branching-on-the-cheap

Search entire site...

About

Documentation

Downloads

GUI Clients
Logos

Community

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

Downloads

 Mac OS X  Windows

 Linux/Unix

Older releases are available and the [Git source repository](#) is on GitHub.

GUI Clients

Git comes with built-in GUI tools (`git-gui`, `gitk`), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).

Latest source Release

2.23.0

Release Notes (2019-08-16)

Download 2.23.0 for Windows





Conta no GitHub



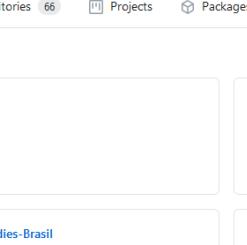
- Neste link
- Caso já tenha, não é preciso criar uma nova conta.
- Se for estudante:
 - [GitHub Student Developer Pack](#) - utiliza email institucional da universidade.

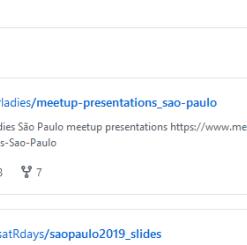


Overview Repositories 66 Projects Packages

Pinned

[cienciadados/dados](#)
Translate Datasets to Portuguese
R 8 ⚡ 8

[rladies/meetup-presentations_sao-paulo](#)
R-Ladies São Paulo meetup presentations <https://www.meetup.com/R-Ladies-Sao-Paulo>
23 ⚡ 7

[satRdays/saopaulo2019_slides](#)
Speaker slides and materials for satRday São Paulo 2019
8 ⚡ 6

[materials_estudo_R](#)
Materiais de estudo de R
58 ⚡ 18

[slidesR](#)
R 1 ⚡ 1

735 contributions in the last year

Less More

Learn how we count contributions.

2020 2019 2018

Mon Tue Wed Thu Fri Sat Sun

Contributed to [beatrizmilz/2020-FLS6397](#).

Activity overview

Code review 2%

Follow ...

191 followers · 185 following · 287

Universidade de São Paulo
São Paulo, Brasil
<https://beatrizmilz.com>
@beamilz

Highlights

Arctic Code Vault Contributor

PRO

Organizations

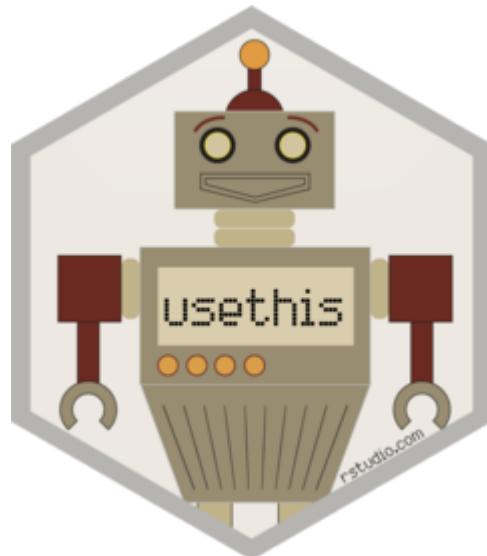


14 / 73

Instalação de pacotes

- Instalar o Pacote `usethis`

```
install.packages("usethis")
library(usethis)
```



Pré-requisitos



Para usar Git na RStudio Cloud

- Conta no GitHub
- Conta no RStudio Cloud

Configurando o Git e GitHub no RStudio

-  No computador: É necessário fazer apenas uma vez (por usuário)!
-  No RStudio Cloud: É necessário fazer a configuração em CADA PROJETO.

Demonstração prática:  Na RStudio Cloud

Configurando o Git e GitHub no RStudio

- Ter o pacote `usethis` instalado
- Se apresentar para o `git`
- Abrir o arquivo `.Renvironment`
- Criar um GitHub Token
- Adicionar o GitHub Token no arquivo `.Renvironment`
- Reiniciar o RStudio



Configurando o Git

- Informar para o Git seu nome, e email.
- Utilize o mesmo email associado à sua conta do GitHub.
- Duas possibilidades: no terminal (linha de comando) ou com uma função do `usethis`.
- Com o `usethis`:

```
usethis::use_git_config(# Seu nome
                        user.name = "Beatriz Milz",
                        # Seu email
                        user.email = "beatriz.milz@hotmail.com")
```



Configurando o Git

- No terminal (Substitua pelo seu nome e email vinculado à conta do GitHub):

```
git config --global user.name "Beatriz Milz"  
git config --global user.email beatriz.milz@hotmail.com
```

```
beatr@DESKTOP-85OP9UB MINGW64 ~  
$ git config --global user.name "Beatriz Milz"  
  
beatr@DESKTOP-85OP9UB MINGW64 ~  
$ git config --global user.email beatriz.milz@hotmail.com
```



Configurando o GitHub + RStudio

- Abra o arquivo `.Renviron` usando a seguinte função:

```
usethis::edit_r_environ()  
# * Edit 'C:/Users/beatr/Documents/.Renviron'  
# * Restart R for changes to take effect
```



Configurando o GitHub + RStudio

- Criar um novo token no GitHub:

```
usethis::browse_github_token()
# ✓ Opening URL 'https://github.com/settings/tokens/new?scopes=repo,gis'
# • Call `usethis::edit_r_environ()` to open '.Renviron'.
# • Store your PAT with a line like:
#   GITHUB_PAT=xxxxyyzzz
#   [Copied to clipboard]
# • Make sure '.Renviron' ends with a newline!
```

- Uma página do GitHub irá abrir, com as configurações necessárias. Você pode alterar o nome do token, para saber em qual computador está vinculando. Crie o novo token.



- Exemplo:

A screenshot of the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins. The main window has tabs for Console, Terminal, and Jobs. The Console tab shows the R startup message and a single character '>'. The Environment tab shows the Global Environment with a list of packages: aMostra-IME-2019-DataVis, doutorado, IME-27-08-2019, NerdzaoWeek, remaster-the-tidyverse, RLadies-Git-RStudio-2019, and Teste-MeetupR. The Files tab shows a directory structure under Home > GitHub with the same folder names. A small yellow circular icon with a white 'I' is positioned near the bottom center of the interface.

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (c) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Caso você esteja vendo a versão em PDF, o video acima não irá funcionar. Veja clicando [neste link](#).



- Copie o token gerado:

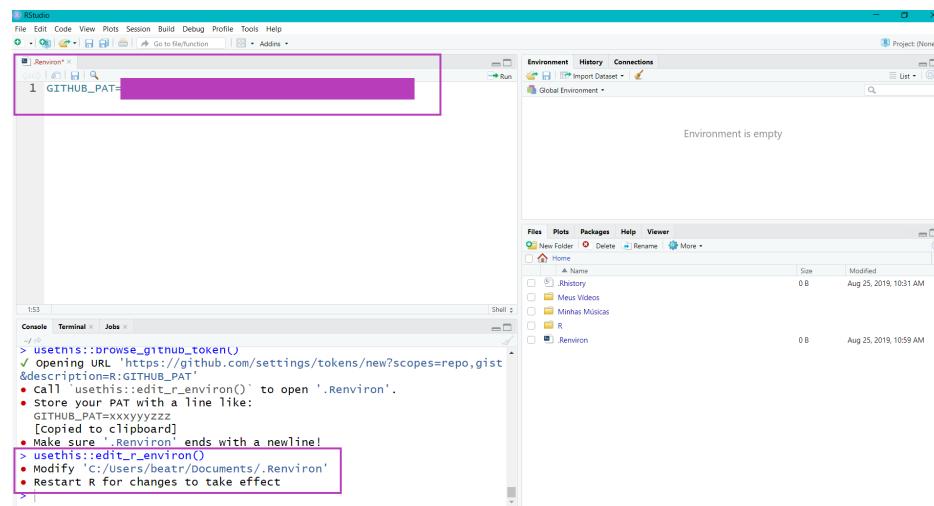
A screenshot of the GitHub Developer settings page, specifically the "Personal access tokens" section. The page has a dark header with navigation links: Pull requests, Issues, Marketplace, and Explore. A message at the top states: "Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved." Below this, the "Personal access tokens" heading is centered, with "Generate new token" and "Revoke all" buttons to its right. On the left, a sidebar lists "GitHub Apps", "OAuth Apps", and "Personal access tokens", with "Personal access tokens" being the active tab. A note below the heading says: "Tokens you have generated that can be used to access the GitHub API." A callout box contains the instruction: "Make sure to copy your new personal access token now. You won't be able to see it again!" Below this, a token entry is shown with a green checkmark, a red "Delete" button, and a blue clipboard icon for copying.



- Abra o arquivo `.Renviron`:

```
usethis::edit_r_environ()
# • Modify 'C:/Users/beatr/Documents/.Renviron'
# • Restart R for changes to take effect
```

- Crie uma nova linha na forma `GITHUB_PAT=SEU_TOKEN`, adicione o token, pule uma linha e salve o arquivo.



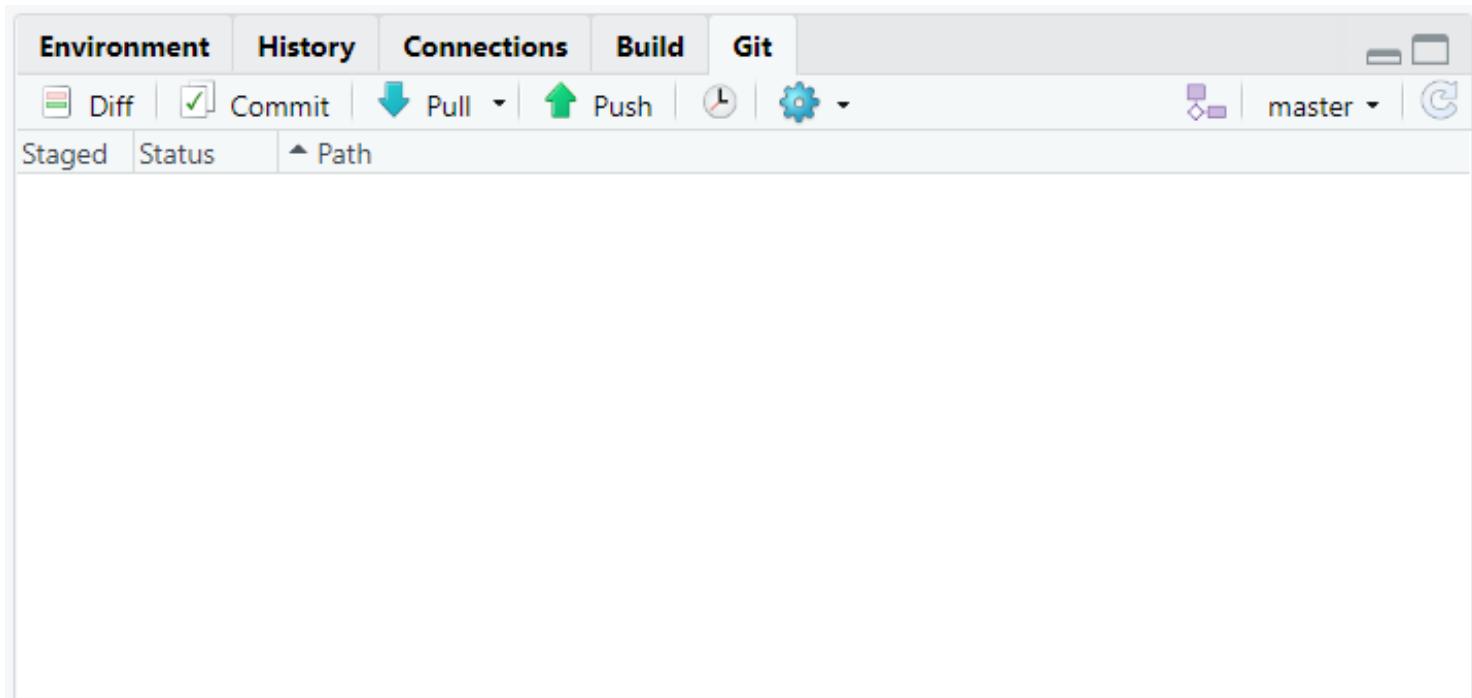
- Reinicie o RStudio: **CTRL + SHIFT + F10**

Interface gráfica do RStudio



Interface Gráfica RStudio

- O RStudio oferece um cliente Git **simples**, na aba "Git" (em inglês, é chamado de Git Pane).
- Esse painel somente aparecerá em projetos que estejam versionados com Git.



Equivalência dos principais comandos



Linha de comando vs RStudio

- git status
- git add [arquivo]
- git commit -m "[mensagem descritiva]"
- git diff
- git pull
- git push
- ...



Git pane - Staged ✓

A screenshot of the RStudio interface focusing on the Git pane. The top navigation bar shows tabs for Environment, History, Connections, Build, and Git. The Git tab is active. Below the tabs is a toolbar with icons for Diff, Commit, Pull, Push, and settings. To the right of the toolbar, it shows the current branch as master. The main area is titled 'Staged' and lists files with their status: M for modified, ? for untracked, and a checkmark for staged. The first two files, 'docs/index.Rmd' and 'docs/index.html', have their checkboxes highlighted with a purple box, indicating they are selected for commit.

Path
docs/index.Rmd
docs/index.html
docs/img/prints/git-pane - Copia.PNG
docs/img/prints/git-pane-branch.png
docs/img/prints/git-pane-diff.PNG
docs/img/prints/git-pane-history.PNG
docs/img/prints/git-pane.PNG

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



Git pane - Commit

The screenshot shows the RStudio interface with the 'Review Changes' tab selected. In the left pane, under the 'Changes' tab, several files are listed as 'Staged' (indicated by blue icons). The top item is 'docs/index.Rmd'. The right pane shows a 'Commit message' input field containing the text 'adiciona slides sobre o git pane'. Below the input field are buttons for 'Amend previous commit' and 'Commit'.

Staged files

The screenshot shows the RStudio interface with the 'Diff' tab selected. The code editor displays a diff between two versions of a file. The changes are color-coded: green for new lines, red for deleted lines, and black for modified lines. The code itself is in R, involving knitr::include_graphics() calls to include images related to the git pane.

```
@@ -610,10 +610,20 @@ class: middle
610 610 # Botão `r emoji::ji("clock3")` (history) -> Review changes
611 611
612 612 ````{r, out.width="90%"}
613 613 knitr::include_graphics("img/prints/git-pane-history.PNG")
614 614 ```

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

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



Git pane - Commit

```
Git Commit Close
>>> c:/Program Files/Git/bin/git.exe commit -F c:/Users/bebe/OneDrive - Documentos/Projetos/PycharmProjects/untitled/commit.txt
[master 8abfe85] adiciona slides sobre o git pane
 2 files changed, 74 insertions(+), 42 deletions(-)
```



Botão Diff -> Review changes

The screenshot shows the RStudio 'Review Changes' interface. The top navigation bar includes 'Changes', 'History', 'master', 'Stage', 'Revert', and 'Ignore'. The main area displays a list of staged files: 'docs/index.Rmd' (M), 'docs/index.html' (M), and 'docs/img/prints/git-pane.PNG' (D). Below this is a commit message field with a 'Commit' button. The bottom half of the window shows a detailed diff view between lines 532 and 537 of a script. Lines 532 and 533 are standard text. Line 534 starts a comment block. Lines 535 and 536 show code for creating a project. Lines 537 and 538 show code for cloning from GitHub, with 'usesthis::create_from_github' highlighted in red and its argument 'destdir' highlighted in green. Lines 539 through 551 show more project creation logic, including ssh key handling and protocol selection, with various lines highlighted in different colors (red, green, pink).

```
@@ -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 usesthis::create_from_github("rstudio-education/remaster-the-tidyverse", #usuário/repositório
+ 537 usesthis::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)
540 540 # 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(usesthis.protocol = "https")`
550 550 #   in your script or in a user- or project-level startup file, '.Rprofile'.
551 551 #   call `usesthis::edit_r_profile()` to open it for editing.
552 552 #   cloning from https://github.com/rstudio-education/remaster-the-tidyverse
```



Botão ⏪ (history) -> Review changes

The screenshot shows the RStudio interface with the 'Review Changes' tab selected. At the top, there's a navigation bar with tabs for 'Changes', 'History', and 'master'. A dropdown menu shows '(all commits)'. On the right, there's a search bar and a pull request icon. Below the navigation is a detailed commit history table:

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	babcbbfe
Beatriz Milz <beatriz.milz@hotmail.com>	2019-08-29	06c6f018

Below the table, a message says 'Commits 1-10 of 10'. The bottom half of the window shows a diff viewer for the file 'docs/index.Rmd'. The diff shows code changes across several lines, with some lines highlighted in different colors (red, green, grey) to indicate different types of changes. The code itself is RMarkdown syntax.

```
@@ -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 ]]
 101 101 .row.bg-main3[.content[
 102 102 ## Configurando o `GitHub`  

 103 103 ]]
 104 104 .row.bg-main4[.content[
 105 105 ## Trabalhando com projetos no `Rstudio` + `GitHub`  

@@ -129,7 +129,7 @@ class: middle
 129 129 # Pré-requisitos
 130 130 
```



Branch

A screenshot of a software interface with a toolbar at the top labeled "Environment", "History", "Connections", "Build", and "Git". Under the "Git" tab, there are buttons for "Diff", "Commit", "Pull", "Push", and a gear icon. Below the toolbar, there are tabs for "Staged" and "Status", and a "Path" dropdown. A dropdown menu for "master" is open, showing options like "Delete", "Rename", and "Switch to branch".

Criar um branch

Verifica em qual branch está
trabalhando
e quais são as branches existentes

Interface Gráfica RStudio

Version Control with Git or SVN



Stage files:

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

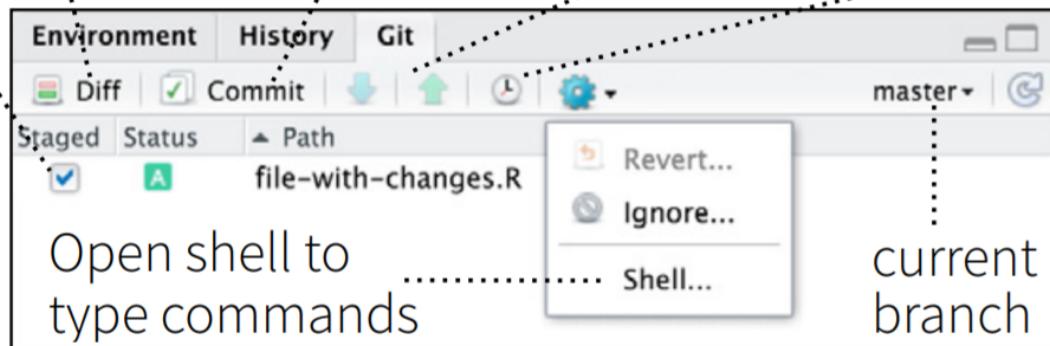
Turn on at **Tools > Project Options > Git/SVN**

Show file diff

Commit staged files

Push/Pull to remote

View History



Fonte: RStudio Cheatsheet

Trabalhando com projetos no RStudio + GitHub



No computador

Método 1 - Clonando um repositório do GitHub



- Crie um repositório no GitHub, ou abra a página de um repositório já existente.
- Criando um repositório no GitHub:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

beatrizmilz / nome-repo

Great repository names are short and memorable. Need inspiration? How about [sturdy-octo-garbanzo](#)?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None Add a license: None

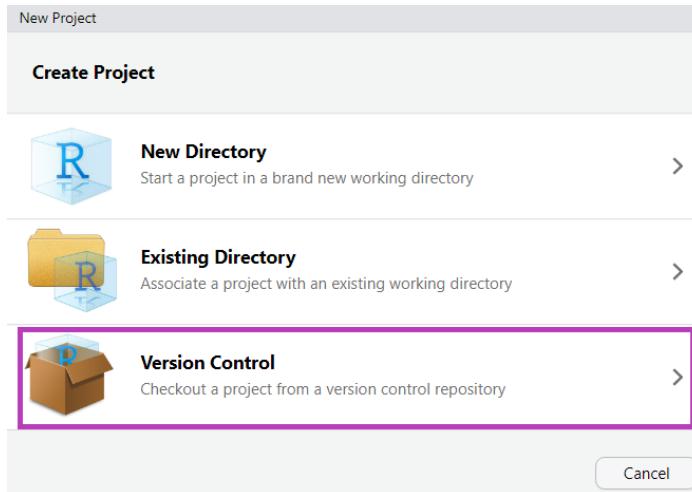
Create repository

Se o repositório não for "seu", faça um fork primeiro, e clone o seu fork.

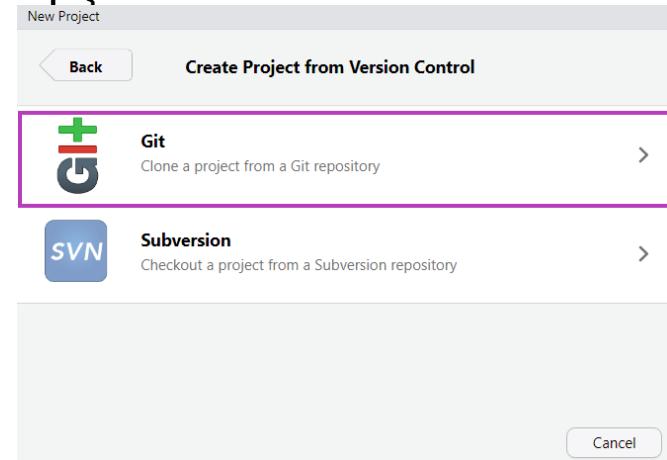


- No RStudio, crie um novo projeto: File > New Project

- Na aba "Create Project", selecione a opção **Version Control**.



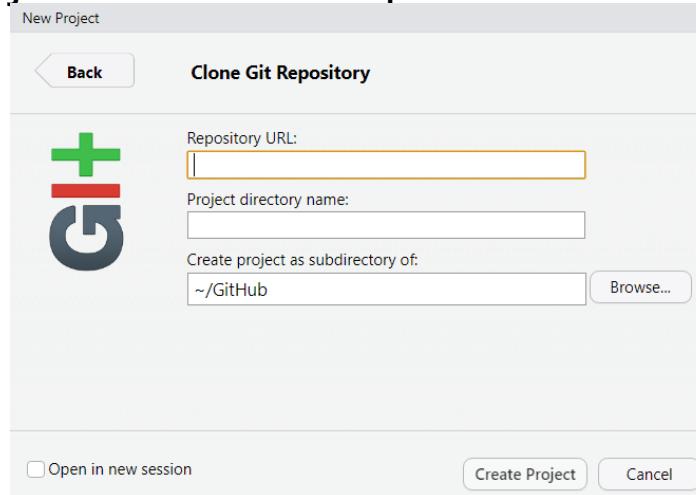
- Na aba "Create Project in Version Control", selecione a opção **Git**.





- Na aba "Clone Git Repository":

- **Repository URL:** Cole o link para o repositório
- **Project directory name:** Após inserir o repository URL, esse campo será preenchido automaticamente.
- **Create project as subdirectory of:** Selecione o diretório onde você deseja manter sua cópia local do repositório.



- O **RStudio** irá fazer o clone do repositório, e abrirá um RProj para ele (caso não exista um ainda, será criado).



Método 2

- **Criando um repositório novo, diretamente do RStudio**
- Vamos usar a função `create_project()`
 - Cria um projeto `.Rproj`
 - Argumento importante: `path` = É o "caminho" para o diretório (pasta). Se o diretório já existe, é utilizado. Se não existe, é criado.
 - Cuidado com o nome do projeto, pois será o mesmo nome que será utilizado no repositório. Você não deve usar o nome de algum repositório já existente no seu GitHub.



Criando um projeto

- No meu caso, tenho o hábito de deixar todos os repositórios do GitHub, na pasta `GitHub/`. No exemplo abaixo, dentro da pasta GitHub, criei o projeto chamado Teste-MeetupR. Esse será o nome da pasta criada e posteriormente do repositório no GitHub.

```
usethis::create_project("GitHub/Teste-MeetupR")
# ✓ Creating 'GitHub/Teste-MeetupR/'
# ✓ Setting active project to 'C:/Users/beatr/Documents/GitHub/Teste-Me
# ✓ Creating 'R/'
# ✓ Writing 'Teste-MeetupR.Rproj'
# ✓ Adding '.Rproj.user' to '.gitignore'
# ✓ Opening 'GitHub/Teste-MeetupR/' in new RStudio session
# ✓ Setting active project to '<no active project>'
```

- O RStudio abrirá outra janela, com o novo projeto criado.



Novo projeto criado!

- Projeto ✓
- Git ✗
- GitHub ✗

A screenshot of the RStudio interface. The title bar says "GitHub/Teste - RStudio". The top menu includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help. The sub-menu under Tools shows "Addins". The main window has tabs for Console, Terminal, and Jobs. The Console tab shows the R startup message:

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"  
Copyright (C) 2019 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
> |
```

The Environment tab shows "Environment is empty". The Global Environment shows no objects. The Files tab shows a directory structure:

Name	Size	Modified
ignore	13 B	Aug 25, 2019, 11:05 AM
R		
Teste.Rproj	407 B	Aug 25, 2019, 11:05 AM



```
usethis::use_git()
# ✓ Setting active project to 'C:/Users/beatr/Documents/GitHub/Te
# ✓ Initialising Git repo
# ✓ Adding '.Rhistory', '.RData' to '.gitignore'
# There are 2 uncommitted files:
# * '.gitignore'
# * 'Teste-MeetupR.Rproj'
# Is it ok to commit them?
#
# 1: Absolutely
# 2: Negative
# 3: No
#
# Selection: 1
# ✓ Adding files
# ✓ Commit with message 'Initial commit'
# • A restart of RStudio is required to activate the Git pane
# Restart now?
#
# 1: No way
# 2: For sure
# 3: Negative
#
# Selection: 2
```

- O RStudio irá reiniciar para que apareça o painel do Git

Usando Git



- Projeto ✓
- Git ✓
- GitHub ✗

A screenshot of the RStudio interface. The title bar shows the path: "/GitHub/Teste · master · RStudio". The top menu includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help. Below the menu is a toolbar with icons for file operations. The main area has tabs for Console, Terminal, and Jobs. The Console tab shows the R startup message:

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"  
Copyright (C) 2019 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

The Environment tab shows "Environment is empty". The Files tab displays the contents of the GitHub repository "Teste":

Name	Size	Modified
.gitignore	32 B	Aug 25, 2019, 11:15 AM
.history	19 B	Aug 25, 2019, 11:15 AM
R		
Teste.Rproj	407 B	Aug 25, 2019, 11:15 AM

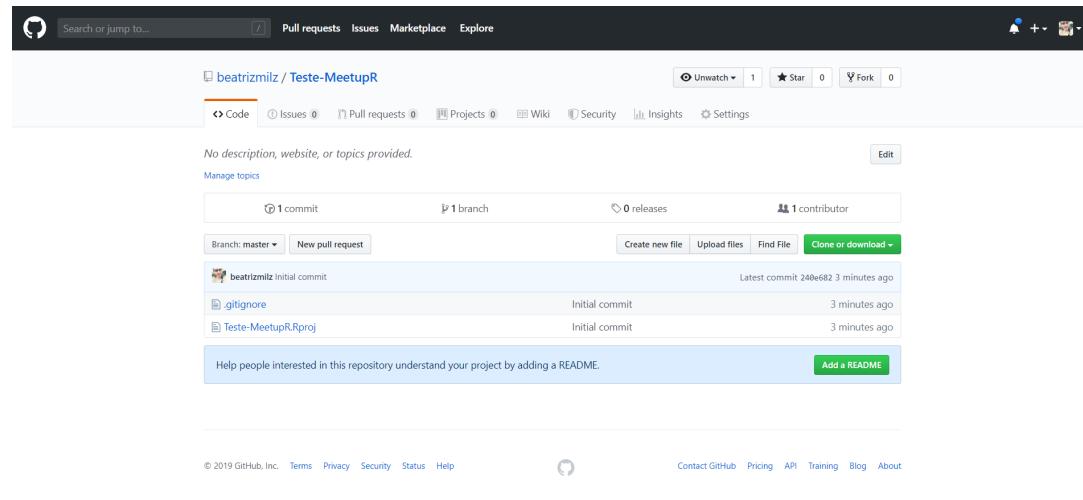


```
usethis::use_github()
# ✓ Setting active project to 'C:/Users/beatr/Documents/GitHub/Te
# ✓ Checking that current branch is 'master'
# Which git protocol to use? (enter 0 to exit)
#
# 1: ssh    <-- presumes that you have set up ssh keys
# 2: https <-- choose this if you don't have ssh keys (or don't know if
#
# Selection: 2
# • Tip: To suppress this menu in future, put
#   `options(usethis.protocol = "https")`
#   in your script or in a user- or project-level startup file, '.Rprof'.
#   Call `usethis::edit_r_profile()` to open it for editing.
# • Check title and description
#   Name:      Teste-MeetupR
#   Description:
# Are title and description ok?
#
# 1: No way
# 2: Definitely
# 3: Nope
#
# Selection: 2
# ✓ Creating GitHub repository
# ✓ Setting remote 'origin' to 'https://github.com/beatrizmilz/Teste-Me
# ✓ Pushing 'master' branch to GitHub and setting remote tracking branc
# ✓ Opening URL 'https://github.com/beatrizmilz/Teste-MeetupR'
```

Repositório criado!



- Projeto ✓
- Git ✓
- GitHub ✓



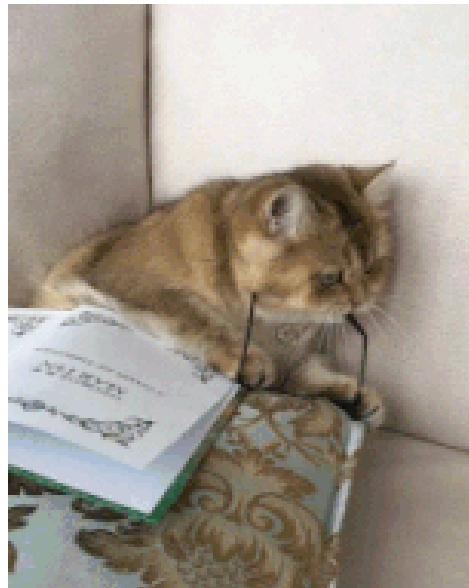
A screenshot of a GitHub repository page for 'beatrilmilz / Teste-MeetupR'. The page shows basic repository statistics: 1 commit, 1 branch, 0 releases, and 1 contributor. It lists two files: '.gitignore' and 'Teste-MeetupR.Rproj'. A green 'Clone or download' button is visible. The GitHub interface includes navigation links like 'Code', 'Issues', 'Pull requests', 'Marketplace', and 'Explore' at the top, and footer links for 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About' at the bottom.



Função para criar um arquivo README.md

```
usethis::use_readme_md()
```

- O arquivo será criado e aberto, para ser editado e salvo.





Exemplo:

A screenshot of the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar below has icons for New Project, Open Project, Save, Run, Stop, and Go to file/function. The main window shows the Environment, History, and Connections panes. The Global Environment pane lists objects. The Files pane shows a directory structure under GitHub, listing folders like aMostra-IME-2019-DataVis, douradodo, IME-27-08-2019, NerdzaoWeek, remaster-the-tidyverse, RLadies-Git-RStudio-2019, and Teste-MeetupR. The bottom pane is a large, empty workspace area.

Caso você esteja vendo a versão em PDF, o video acima não irá funcionar. Veja clicando [neste link](#).



Método 3

- **A partir de um repositório existente, fazer um fork e clonar diretamente do RStudio**
- Para isso, usaremos a função `usethis::create_from_github()`.
- **Não use** essa função no RStudio Cloud.
- Argumentos importantes:
 - `repo_spec` = repositório que quer trabalhar. Deve estar no formato: "usuario_github/repositorio". Veja o exemplo abaixo.
 - `destdir` = diretório onde quer que os arquivos sejam salvos
 - `fork` = se igual a TRUE, irá fazer o fork do repositório. Só funcionará se o token do GitHub foi configurado corretamente.



Exemplo do uso da função `create_from_github()`

- Exemplo: fork e clone do projeto [Lista de meetups R-Ladies no Brasil](#):

```
usethis::create_from_github("R-Ladies-Sao-Paulo/RLadies-Brasil",
                           destdir = "C:/Users/beatr/Documents/GitHub",
                           fork = TRUE)
# ✓ Creating 'C:/Users/beatr/Documents/GitHub/datascience-box/'
# ✓ Forking 'R-Ladies-Sao-Paulo/RLadies-Brasil'
# ✓ Cloning repo from 'https://github.com/beatrizmilz/RLadies-Brasil.git'
# ✓ Setting active project to 'C:/Users/beatr/Documents/GitHub/RLadies-Brasil'
# ✓ Adding 'upstream' remote: 'https://github.com/R-Ladies-Sao-Paulo/RLadies-Brasil.git'
# ✓ Pulling changes from GitHub source repo 'upstream/master'
# ✓ Setting remote tracking branch for local 'master' branch to 'upstream/master'
# ✓ Opening 'C:/Users/beatr/Documents/GitHub/RLadies-Brasil/' in new RStudio
# ✓ Setting active project to 'C:/Users/beatr/Documents/GitHub/RLadies-Brasil'
```



Crie uma branch

- Antes de fazer qualquer modificação, crie uma branch, usando a função `usethis::pr_init()`.
- Argumento importante: `branch` = indica o nome da branch que você quer criar.

```
usethis::pr_init(branch = "teste")
```

```
# ✓ Checking that local branch 'master' has the changes in 'origin/mast
# ✓ Creating local PR branch 'teste'
# ✓ Switching to branch 'teste'
# • Use `pr_push()` to create PR
```



Faça alterações e commits

- Faça suas contribuições, e sempre faça commits com mensagens que deixem claro as mudanças.
- Quando terminar as contribuições necessárias, utilizar a função `pr_push()`:

```
pr_push()  
# ✓ Pushing local 'teste' branch to 'origin:teste'  
# ✓ Setting upstream tracking branch for 'teste' to 'origin/teste'  
# ✓ Create PR at link given below  
# ✓ Opening URL 'https://github.com/beatrizmilz/RLadies-Git-RStudio-201'
```

- A página do GitHub será aberta, para que possa completar o Pull Request.



Crie um Pull Request:

The screenshot shows a GitHub repository page for 'beatrizmilz / RLadies-Git-RStudio-2019'. The main heading is 'Comparing changes' with a note: 'Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.' Below this, a button says 'Create pull request'.

Branches compared: base: master vs compare: teste. A message indicates they are 'Able to merge'. The commit details show 1 commit, 1 file changed, 0 commit comments, and 1 contributor (beatrizmilz). The commit message is 'Adiciona função pr_push()' and the commit hash is d876c2f. It was made on Sep 01, 2019.

Below the commit details, it says 'Showing 1 changed file with 14 additions and 3 deletions.' The file shown is 'docs/index.Rmd'. The diff view highlights changes in the code, such as the addition of a branch argument in the 'usethis::pr_init' function call.

Line	Code	Comment
603	usethis::pr_init(branch = "teste")	- Argumento importante: `branch` indica o nome da branch que você quer criar.
604		604
605	usethis::pr_init(branch = "teste")	605
606	+usethis::pr_init(branch = "teste")	+usethis::pr_init(branch = "teste")
607		607



Open a Pull Request:

A screenshot of a GitHub repository page for 'beatrizmilz / RLadies-Git-RStudio-2019'. The top navigation bar shows 'Code' is selected, along with 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Security', 'Insights', and 'Settings'. The pull request count is 0. Below the navigation, the title 'Open a pull request' is displayed, followed by the instruction 'Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.' A dropdown menu shows 'base: master' and 'compare: teste', with a note that they are 'Able to merge'. The main area contains a text editor with the title 'Demonstração' and the content 'Demonstração feita para prints na apresentação. Foi adicionado conteúdo sobre a função pr_push()'. A purple arrow points from the text 'Foi adicionado conteúdo sobre a função pr_push()' to the merge status message. To the right, there are sections for 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), and 'Milestone' (No milestone). A large green 'Create pull request' button is at the bottom right of the editor.



Exemplo: PR foi aberto

The screenshot shows a GitHub repository page for 'beatrilmilz / RLadies-Git-RStudio-2019'. The 'Pull requests' tab is selected, showing one open pull request. The pull request details are as follows:

- Open**: beatrilmilz wants to merge 2 commits into `master` from `teste`.
- Conversation**: 0 comments.
- Commits**: 2 (by beatrilmilz):
 - `Adiciona função pr_push()` (commit hash: d876c2f)
 - `adiciona imagem pr-push` (commit hash: c582fb1)
- Checks**: 0
- Files changed**: 2
- Reviewers**: No reviews
- Assignees**: No one—assign yourself
- Labels**: None yet
- Projects**: None yet

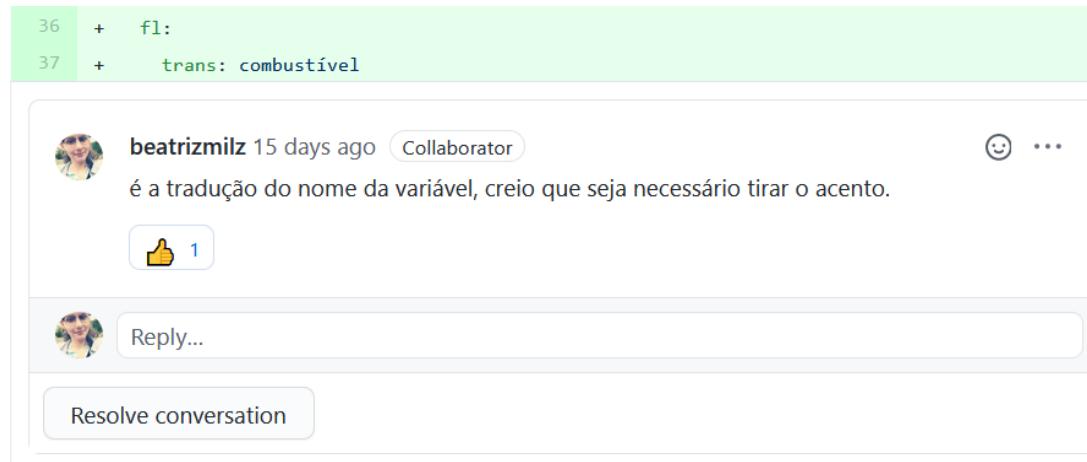
A comment from beatrilmilz is highlighted:
beatrilmilz commented now
Demonstração feita para prints na apresentação.
Foi adicionado conteúdo sobre a função `pr_push()`.

- Depois dessa etapa, você deve esperar a pessoa que mantém o repositório responder fazendo sugestões para melhorias, aceitando seu PR, etc.



Modificar o PR

- Alguém fez uma sugestão, como alterar o PR enviado? Ex:



A screenshot of a GitHub pull request interface. At the top, two code snippets are shown:

```
36 + fl:  
37 + trans: combustível
```

Below this, a comment from user `beatrizmilz` is displayed, dated 15 days ago and marked as a collaborator. The comment text is:

é a tradução do nome da variável, creio que seja necessário tirar o acento.

Underneath the comment, there is a blue thumbs-up icon with the number '1' next to it, indicating one like. Below the comment area is a 'Reply...' button and a 'Resolve conversation' button.

- No mesmo projeto e na **mesma branch** que você usou para enviar o PR, faça as alterações.
- Selecione o arquivo modificado para deixá-lo disponível para criar um Commit. Faça o commit.
- Para enviar o novo commit, use o Git Push (aperte a seta verde para cima no painel do Git do RStudio).



Depois que o PR é aceito: Merged

Screenshot of a GitHub pull request merge page for the repository `beatrilmilz / RLadies-Git-RStudio-2019`.

The pull request has been merged into the `master` branch from the `teste` branch 3 minutes ago.

Comments:

- beatrilmilz** commented 4 minutes ago:
Demonstração feita para prints na apresentação.
Foi adicionado conteúdo sobre a função `pr_push()`.
- beatrilmilz** added 2 commits 9 minutes ago:
 - Adiciona função `pr_push()` (commit `d876c2f`)
 - adiciona imagem `pr-push` (commit `c582fb1`)
- beatrilmilz** merged commit `5cf7727` into `master` 3 minutes ago

Status message:

Pull request successfully merged and closed
You're all set—the `teste` branch can be safely deleted.

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Customize

Unsubscribe



Última etapa: APÓS o PR ter sido aceito (merged)

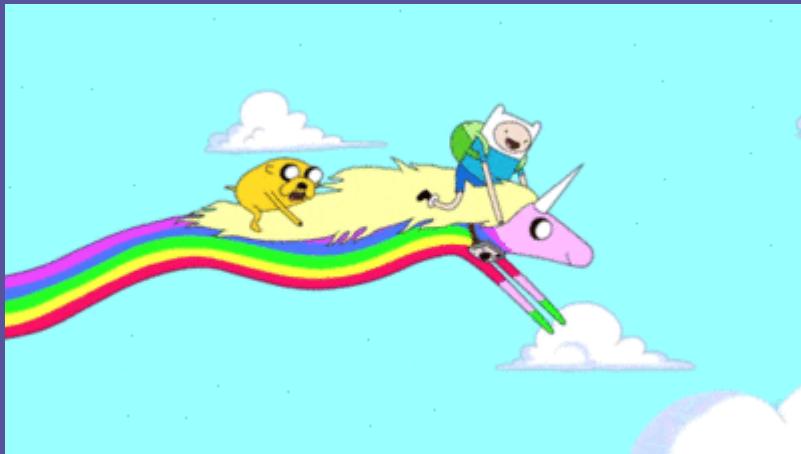
- A função `pr_finish()`:
 - Retorna para master
 - Atualiza (usando o pull) o conteúdo
 - Deleta a branch criada para enviar o PR

```
pr_finish()  
# ✓ Switching back to 'master' branch  
# ✓ Pulling changes from GitHub source repo 'origin/master'  
# ✓ Deleting local 'teste' branch
```

- Pronto!

Trabalhando com projetos no RStudio + GitHub

Cloud Na RStudio Cloud



Trabalhando na RStudio Cloud



- Não é a única forma, é a forma que eu achei mais simples.

Etapa 1:

- Escolha um repositório **seu** para clonar. Pode ser um novo repositório, ou um já existente.

- Lembrando, caso queira criar um repositório no GitHub:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *



beatrizmilz ▾

Repository name *

nome-repo



Great repository names are short and memorable. Need inspiration? How about [sturdy-octo-garbanzo](#)?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: None ▾



Create repository



Etapa 2:

- No RStudio Cloud, Clique na seta ao lado de 'New Project' e escolha a opção "New Project from Git Repo"

The screenshot shows the RStudio Cloud interface. At the top, there are navigation tabs: 'Your Workspace' (selected), 'Projects' (highlighted in blue), and 'Info'. On the far right, there's a user profile icon for 'Beatriz Milz' and a trash bin icon. Below the tabs, the 'Your Projects' section lists four projects: 'amostra-listas', 'curriculo_no_R', 'Curso_Basico_de_R', and 'estrutura_rmd'. Each project entry includes a 'Delete' button and a 'Move' button. To the right of the project list is a modal window titled 'Options'. Inside the modal, there's a 'Search Projects' input field with a magnifying glass icon. Below it is a 'Sort Projects' section with two radio buttons: 'By name' (selected) and 'By date created'. At the bottom of the modal, there's a note: 'Capacity This is your personal workspace.' and a link: 'Learn more about Your Workspace in the Guide.'

1 New Project

2 New Project from Git Repo



Etapa 2: continuação

- Indique a URL do repositório (o link) e clique "OK". O RStudio Cloud criará um projeto onde esse repositório será clonado.

The screenshot shows the RStudio Cloud interface. On the left, there's a sidebar with 'Your Projects' containing four entries: 'amostra-listas', 'curriculo_no_R', 'Curso_Basico_de_R', and 'estrutura_rmd'. The 'Projects' tab is selected at the top. In the center, a modal dialog box titled 'New Project from GitHub Repository' is open. It has a text input field labeled 'URL of your GitHub repository' with a placeholder 'https://github.com/'. Below the input field is a blue 'OK' button. A large, semi-transparent text overlay in pink reads 'Escreva a URL do seu repositório (e depois clique 'OK')' with two arrows pointing towards the input field and the 'OK' button respectively. On the right side of the screen, there's an 'Options' sidebar with sections for 'Search Projects', 'Sort Projects' (with radio buttons for 'By name' and 'By date created' selected), and 'Capacity' (noting it's a personal workspace).



Etapa 3:

- Se apresente ao Git! O melhor, neste caso, é via terminal.

The screenshot shows the RStudio interface. On the left, the code editor displays a Markdown file named 'README.md' containing information about three international seminars. On the right, the 'Files' panel shows the project structure with files like '.gitignore', '.Rhistory', 'project.Rproj', and 'README.md'. At the bottom, a terminal window is open in the 'Terminal' tab, showing the command 'rstudio-user@application-2697924-deployment-7003350-cw97:/cloud/project\$'.

```
5 ##### Eventos recorrentes:
6
7 ##### International Seminar on Statistics with R
8 - Época: 1º semestre
9 - Local: Niterói, Rio de Janeiro, Brasil
10 - Website: http://ser.uff.br/
11
12 ##### RDay
13 - Época: 2º Semestre
14 - Local: Curitiba, Paraná, Brasil
15 - Website: http://rday.leg.ufpr.br/
16
17 ##### LatinR
18 - Época: 2º semestre
19 - Local: Não é fixo.
20 - Website: http://latin-r.com/
21
```

.... se apresente
aqui :)



Etapa 3: continuação

- Se apresente ao Git!
- No terminal (Substitua pelo seu nome e email vinculado à conta do GitHub):

```
git config --global user.name "Beatriz Milz"
```

```
git config --global user.email beatriz.milz@hotmail.com
```

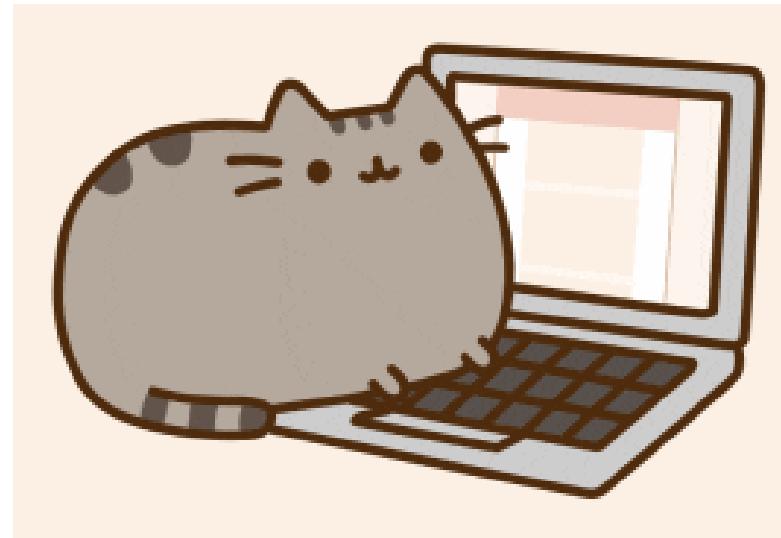


Etapa 4

- Faça as mudanças no código!
- Quando quiser, faça o commit das mudanças.
- Para enviar as mudanças locais para o repositório remoto, faça o Push. Na primeira vez que for fazer isso, vai abrir uma janela e pedir seu usuário e senha do GitHub (para autenticação).
- Após o push, confira no GitHub se as alterações que você fez estão lá :)

Linha de comando ou interface gráfica?

Use o que for melhor para você!



Confira a cheatsheet do Git!



Preparades para a Hacktoberfest 2020?

<https://hacktoberfest.digitalocean.com/>





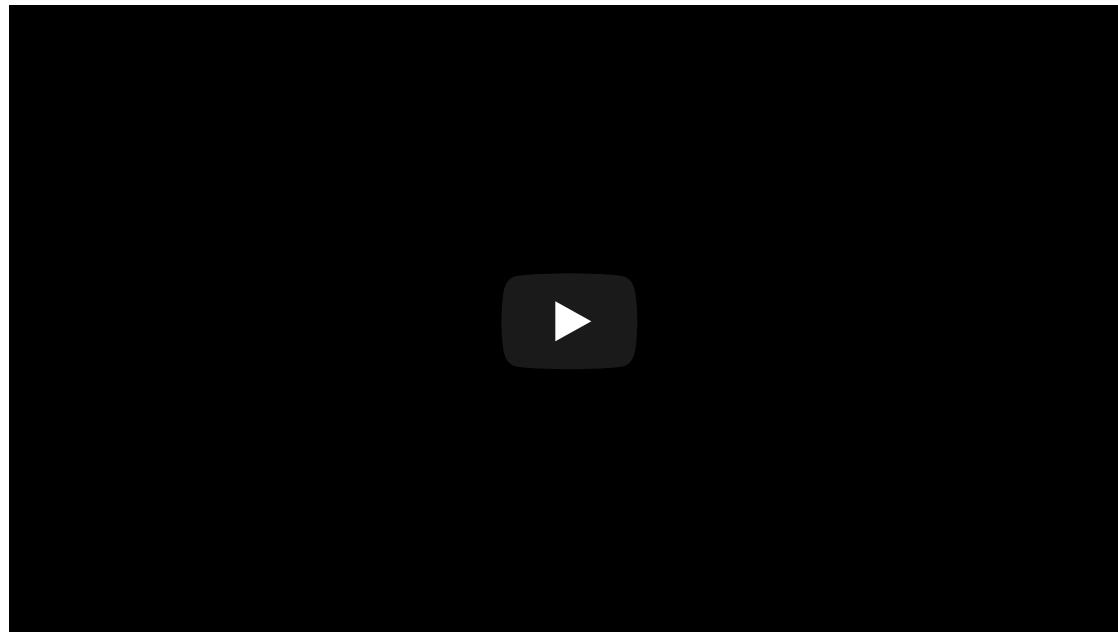
Referências

- Parte deste material foi baseado/inspirado neste post do blog da [Curso-R](#), escrito por [Caio Lente](#):
 - [Git e GitHub \(Zen do R - Parte 4\)](#)
- Outra referência importante: [Happy Git and GitHub for the useR - Jenny Bryan](#)
- Documentação do pacote [usethis](#)



Referências

Palestra (em inglês) da Jenny Bryan:





R-Ladies no Brasil

- Quer saber mais sobre próximos eventos das R-Ladies por aqui? Favorite este [repositório no GitHub](#) ❤️:



Capítulos e eventos online da R-Ladies no Brasil

18



O que achou da apresentação?

- Por favor, responda o formulário abaixo, pois me ajuda **muito** a melhorar os materiais e as apresentações ❤️:

O que achou da apresentação?

Te convido a responder este rápido questionário sobre a sua opinião em relação à apresentação que você participou.

Todas as respostas são importantes para que o material e a explicação possa ser aprimorada!

Muito obrigada.

*Obrigatório

Selecione abaixo qual evento você participou hoje: *

Escolher



Obrigada!

Slides criados com o pacote **xaringan**.

Tema criado com o pacote **xaringanthemer** e funções extras com **xaringanExtra** e **countdown**.

Várias ilustrações usadas na apresentação foram feitas por [Allison Horst](#). Clique [aqui](#) para ver várias outras artes feitas por ela!

O maravilhoso logo da R-Ladies usado nesta apresentação é uma obra de [Bea @Chucheria](#)! Obrigada!

- Caso encontre algum erro nesta apresentação, [escreva uma issue neste repositório](#) para que o material seja aprimorado.