

Gerenciamento de projetos com RStudio, Git e Github

Willber Nascimento

20 de maio de 2022

FACEP e PPGCP/UFPE

Projetos?

O que é projeto?

“Um projeto na área de negócios e na ciência é normalmente definido como um empreendimento (seja ele colaborativo ou não), frequentemente envolvendo **pesquisa** ou **desenho**, que tem como objetivo alcançar um **resultado exclusivo** (Wikipédia).”

Um projeto de pesquisa envolve algumas dessas condições

- Objetivo único
- Empreendimento, múltiplas etapas
- Específico: referências e dados próprios

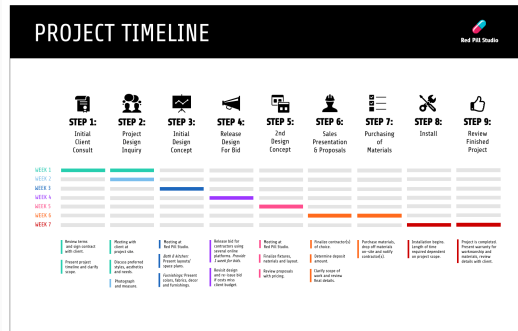


Figura 1: Projeto

Fonte: Google Images

O projeto na análise de dados

Na análise de dados e na CP precisamos garantir que projetos sejam **replicáveis**

- Reprodutível: necessário pouco conhecimento sobre o trabalho para obter os resultados
- Autocontido: o projeto não depende de ferramentas/dados externos à ele
- Documentado: código comentado e codebooks disponíveis e acessíveis



Figura 2: Acessibilidade

Fonte: @PolSciReplicate

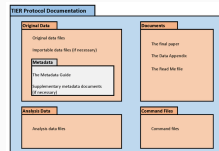
O projeto na comunidade acadêmica

A comunidade acadêmica produz padrões de comportamento



(a) OSF

- Open Science Framework
- Protocolo Tier
- DataVerse



(b) TIER



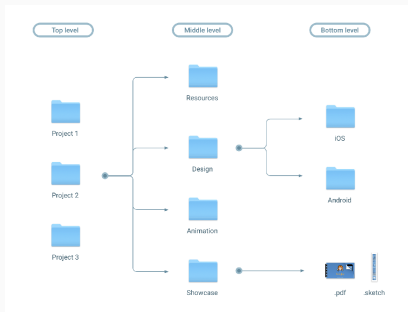
(c) DataVerse

Figura 3: Protocolos e plataformas de transparência

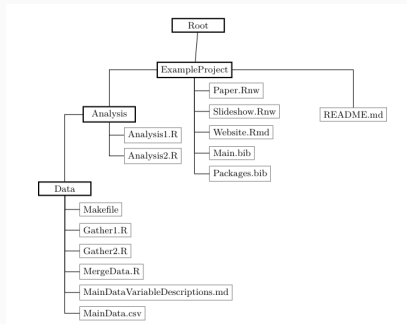
O projeto na prática

Um projeto, na prática é uma pasta no sistema de arquivos do computador

- Uma pasta para cada projeto.



(a) Estrutura genérica, Fonte: Google



(b) Gandrund(2017)

- Cada projeto possui sua estrutura e arquivos
- Os scripts são mais importantes

O projeto na prática II: RStudio

Um ambiente de desenvolvimento integrado (IDE) facilita a configuração de projetos

- Não é obrigatório, mas ele ajuda
- Possui muitos recursos:
 - Criação de projetos por tipo: análise de dados, pacotes, etc.
 - Localização da raiz do projeto (path): chega de setwd na linha 1
 - Integração com controles de versão
 - Automatização de build
 - Execução em paralelo
 - E muitas outras: ver em Rstudio CheatSheet.

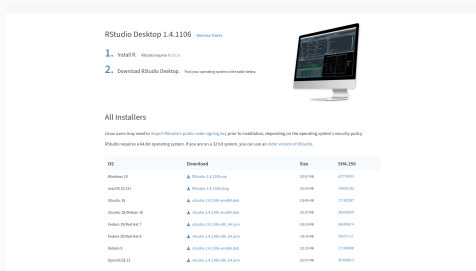


Figura 5: Projetos com Rstudio, Fonte: Google.

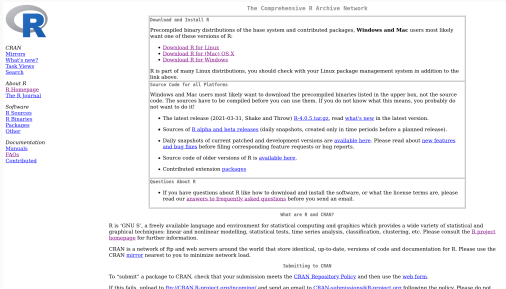
O projeto na prática II: RStudio uma demonstração

- Abrir Rstudio
- Criar um projeto, verificar o path do diretório de trabalho
- Criar diretórios, mudar de path: setwd, getwd, paths absolutos vs relativos
- Opções de configuração do Rstudio (Tools -> Global Options)

Instalação e configuração dos pacotes

Você vai precisar instalar

- R: usuários de Win, opcionalmente, devem instalar o RTools.
- Rstudio



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 list, 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 (2021-03-31, Shaka and Throat) [R-4.0.3.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 [source snapshots 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 tools, time series analysis, classification, clustering, etc. Please consult the [R package 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.

Submitting to CRAN

To "submit" a package to CRAN, check that your submission meets the [CRAN Repository Policy](#) and then use the [upload form](#).

If this fails, submit to [ftp://R-project.org/pub/contrib](#) and send an email to CRAN-submission@R-project.org following the rules. Please do not

(a) Página do CRAN

RStudio Desktop 1.4.1106 - Release Notes

1. Install R. [RStudio requires R 3.5.1+.](#)
2. Download RStudio Desktop. Find your operating system in the table below.



All Installers

Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy. RStudio requires a 64-bit operating system. If you are on a 32-bit system, you can use an older version of RStudio.

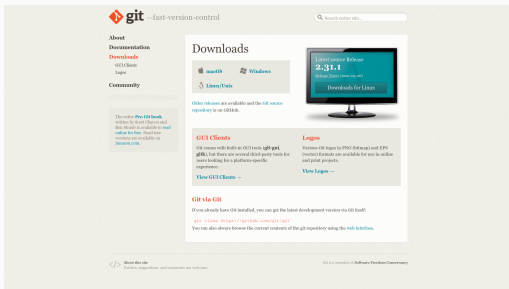
OS	Download	Size	SHA-256
Windows 10	RStudio-1.4.1106.exe	255.01 MB	45111601
macOS 10.13+	RStudio-1.4.1106.dmg	255.01 MB	0882026a
Ubuntu 18	RStudio-1.4.1106-amd64.deb	136.01 MB	17102087
Ubuntu 18/Debian 10	RStudio-1.4.1106-amd64.deb	131.07 MB	30505925
Fedora 15/Red Hat 7	RStudio-1.4.1106-x86_64.rpm	136.10 MB	49848044
Fedora 26/Red Hat 8	RStudio-1.4.1106-x86_64.rpm	136.10 MB	3046763a
Debian 9	RStudio-1.4.1106-amd64.deb	131.23 MB	07103088
OpenSUSE 15	RStudio-1.4.1106-x86_64.rpm	133.07 MB	35308823

(b) Página do Rstudio

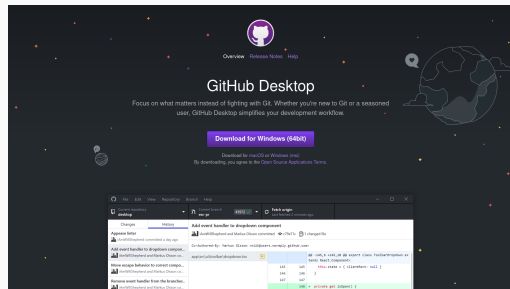
Git e Github: download

Você vai precisar baixar e instalar

- Git: identifique a versão para seu sistema operacional, baixe e instale.
- Github: Baixe a versão para seu SO, mas não há cliente para Linux.
- Existem outros serviços além do github: vejam aqui



(a) Git



(b) Github

Git: configuração básica

- Abra o git shell, powershell, terminal, ou o terminal dentro do Rstudio e execute

```
$ git config --global user.name "Fulano de Tal"
```

```
$ git config --global user.email "fulanodetal@exemplo.br"
```

- Teste sua configuração com:

```
$ git config --list
```

- Seu e-mail precisa ser aquele que vamos usar no Github
- Veja mais informações na documentação oficial

Vamos criar uma conta no site do Github

- Acesse a página de login e crie sua conta
- Seu e-mail precisa ser aquele que configuramos no Git no pc
- Veja outras opções ao Github: (a) Gitlab, (b) Bitbucket

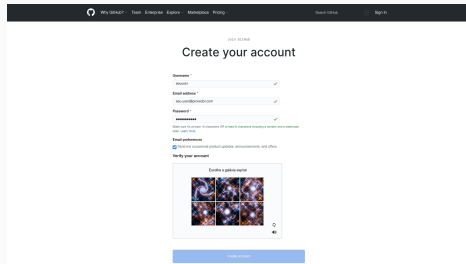


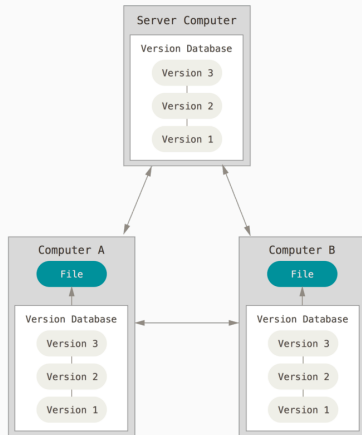
Figura 8: Criar conta no Github

Controle de versão com Git e Github

Git: conceitos básicos do controle de versão

“Controle de versão é um sistema que **registra alterações** em um arquivo ou conjunto de arquivos ao longo do tempo para que você possa **lembrar versões** específicas mais tarde(Shacon e Straub 2009).”

- O registro de alterações com base em versão
- Funciona muito bem com arquivos não-binários
- Reverter, comparar mudanças e recuperar arquivos
- Colaboração com colegas: todos compartilham o projeto



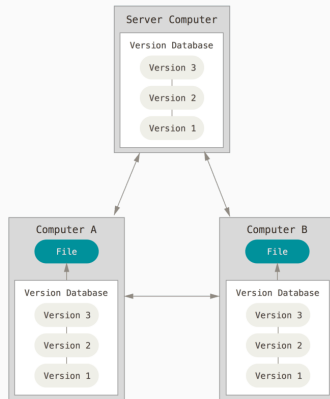
Git: iniciando e clonando um projeto

O Git gerencia seu projeto, por isso ele precisa ser iniciado dentro dele. Quando você copia o projeto de alguém que usa git, isso já vem feito

- Para iniciar o git execute dentro do projeto: `git init`
- Para copiar um projeto use: `git clone https://projeto.git`
- Vamos vamos para a prática!

Lista 1 - Inicializando o git e clonando projetos

- Crie um projeto e inicie o git dentro dele
- Clone um projeto do github



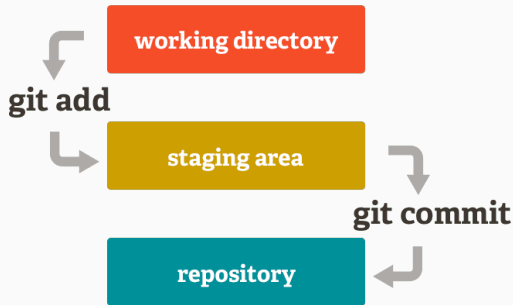
Git: Trabalhando com a área de stage (status, add, commit)

É um local/estado temporário que mantém seus arquivos em edição. Pense que você tem versões passadas salvas, a versão atual (working directory) e edições atuais.

- `git status` apresenta o estado da área
- `git add` para adicionar arquivos ao stage
- `git commit` salva suas alterações com um rótulo

Lista 2 - Básico da area stage

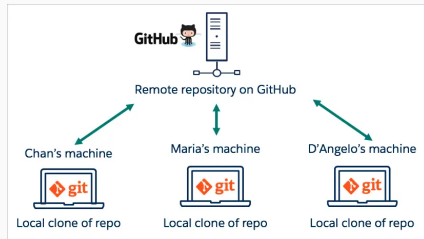
- Use `git status`, crie um arquivo e visualize o staged novamente
- Use o `git add` para adicionar o arquivo ao stage
- Use o `git commit` para registrar a alteração



Git: Sincronizando projeto remoto

O seu repositório no gitlab, github ou outros é conhecido como repositório remoto.

- Cópia do projeto local com as alterações sincronizadas (ele é de fato o main)
- Necessário ter uma conta em um dos serviços
- Linkar a cópia local com diretório remoto
- Usar `git push` para enpuurar as mudanças
- Usar `git pull` para puxar as mudanças
- Veja mais informações sobre os repositórios com `git remote`



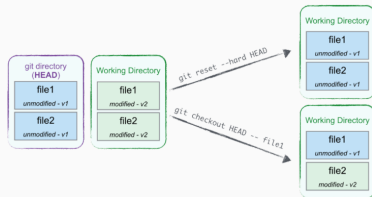
Lista 3 - Repositório remoto

- Sincronize repositório local e remoto
- Use o `git push` para enviar alterações
- Use o `git pull` para puxar alterações do remoto

Git: resetando mudanças

Quase sempre você pode desfazer alterações em seus arquivos. O git entrega muitas ferramentas para cumprir a tarefa, mas o *reset* e *checkout* são bastante comuns.

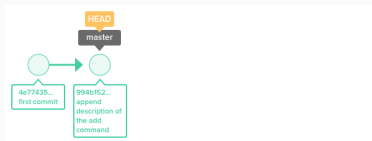
- Muitas vezes criamos um commit de um arquivo antes o tempo: podemos usar o `git reset` para remover o arquivo do staging.
- Quando queremos descartar as mudanças feitas em um arquivo usamos o `git checkout --file` para apagar alterações.
- Se a ideia é editar a mensagem de commit anterior ou adicionar arquivos a ele `git commit --amend`



(a) reset e checkout

Lista 4 - Removendo alterações

- Resete o último commit (`git reset --soft HEAD~1`)
- Remova as alterações de um arquivo (`git checkout -- file`)
- Reescreva a mensagem de um commit (`git commit --amend`)



(b) amend

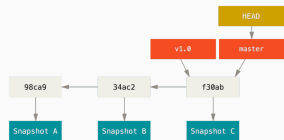
Git: usando branches para desenvolvimento

Com o git você consegue bifurcar seu projeto em *branches*/ramos. Esses ramos são cópias (snapshots) isoladas do seu projeto em um dado momento (commit).

- Todo projeto inicia na branch principal (master/main)
- As use para desenvolver/testar funcionalidades antes de integrá-las
- Ótimo para gerenciar contribuições de colegas

Lista 5 - Lindando com branches

- Crie uma branch (`git branch nomeBranch`)
- Mude para a nova branch (`git checkout nomeBranch`)
- Você pode enviar a nova branch para o remoto (`git --set-upstream origin nomeBranch`)



(a) Commits como snapshots



(b) Branch master e testing

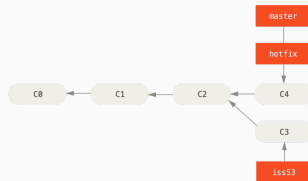
Git: merge de branches

Uma vez que você tem a funcionalidade finalizada na sua *branch* de desenvolvimento é possível integrá-la no ramo master.

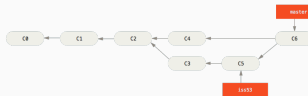
- Sua branch de dev está em momentos diferentes do master
- Esse procedimento é chamado de merge
- Existem muitos tipos de merge: os mais simples é fast-forward

Lista 6 - Merging com branches

- Mude para a branch master (`git checkout master`)
- Integre as mudanças na master (`git merge nomeBranch`)



(a) Branch dev e master

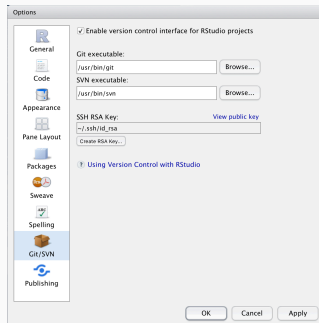


(b) Merge dev com master

Integração do R/RStudio com Git e Github

O Rstudio facilita o uso do git disponibilizando ferramentas gráficas para integração.

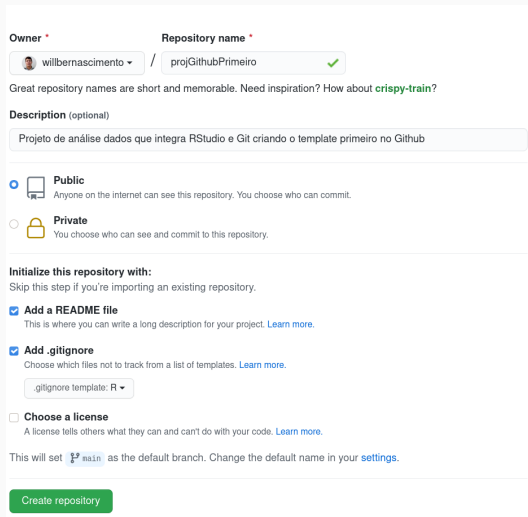
- Tools -> Global Options -> Git/SVN -> Enable Version Control
- O fluxo de trabalho baseado em projetos se integra facilmente
- Existem multiplos cenários para inicialização:
 - Um projeto novo Github + Rstudio
 - Um projeto novo Rstudio + git + github



Rstudio/Git: Iniciando um projeto no Github primeiro


A maneira simples de iniciar um projeto que integra Git + Rstudio é criar primeiro o projeto no Github (repositório remoto)

- Cria-se um repositório no Github/Lab
- Clona-se o repositório no computador
- Inicia-se um projeto RStudio dentro dele



The screenshot shows the GitHub 'Create repository' form. At the top, the 'Owner' is 'willbernascimento' and the 'Repository name' is 'projGithubPrimeiro', which is marked with a green checkmark. Below this, a tip suggests repository names should be short and memorable, with an example 'crispy-train?'. The 'Description' field contains the text 'Projeto de análise dados que integra RStudio e Git criando o template primeiro no Github'. Under the 'Visibility' section, 'Public' is selected, indicating that anyone on the internet can see the repository. The 'Initialize this repository with' section has two checked options: 'Add a README file' and 'Add .gitignore'. The 'Choose a license' option is unchecked. At the bottom, it states the default branch will be 'main' and provides a link to 'settings'. A green 'Create repository' button is at the very bottom.


Owner * Repository name *


 willbernascimento / projGithubPrimeiro ✓

Great repository names are short and memorable. Need inspiration? How about [crispy-train?](#)

Description (optional)

Projeto de análise dados que integra RStudio e Git criando o template primeiro no Github

☒  **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.


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

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

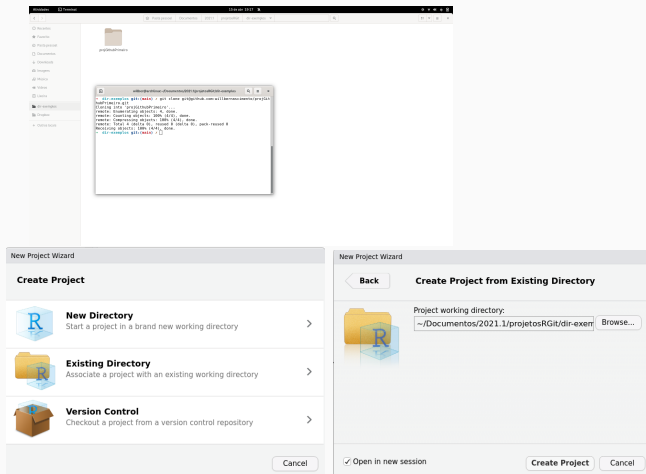
.gitignore template: R ▾

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

Create repository

Rstudio/Git: Iniciando um projeto no Github primeiro



Lista 7 - Crie um projeto no github + Rstudio

- Faça o clone no local desejado
- Crie um novo projeto no Rstudio -> na pasta do clone
- Abra na nova seção

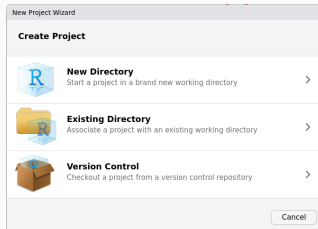
Rstudio/Git: Um projeto novo Rstudio + git + github

Neste cenário usaremos um novo projeto criado no RStudio e configuraremos as demais etapas.

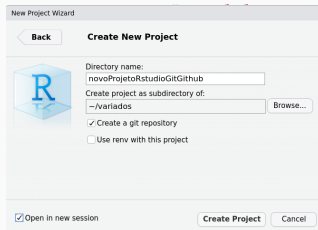
- Criar um projeto no RStudio
- Marcar opção de novo diretório
- Marcar opção de controle de versão git
- Criar o remoto no github e adicioná-lo no projeto

Lista 8 - Criar projeto integrado

- Crie um novo projeto iniciando no Rstudio + Git
- Configure um repositório remoto github neste projeto



(a) diretório novo



(b) criar git

Rstudio/Git: Um projeto novo Rstudio + git + github

- Quando criamos nosso projeto local primeiro, queremos dar um push
- Então adicionamos o remoto, adicionamos arquivos e commits
- Finalmente usamos o push

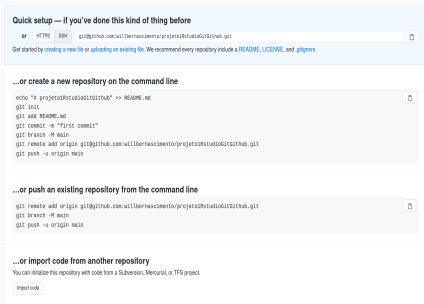


Figura 15: comandos github

O Github ensina procedimentos diferentes para cada tipo de setup.

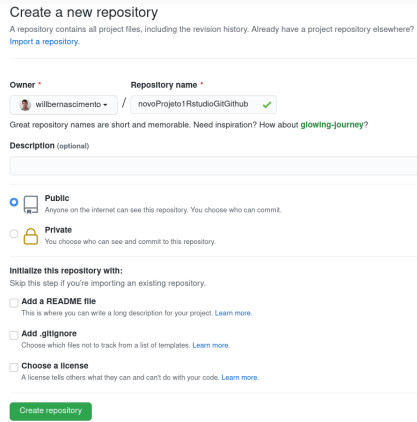
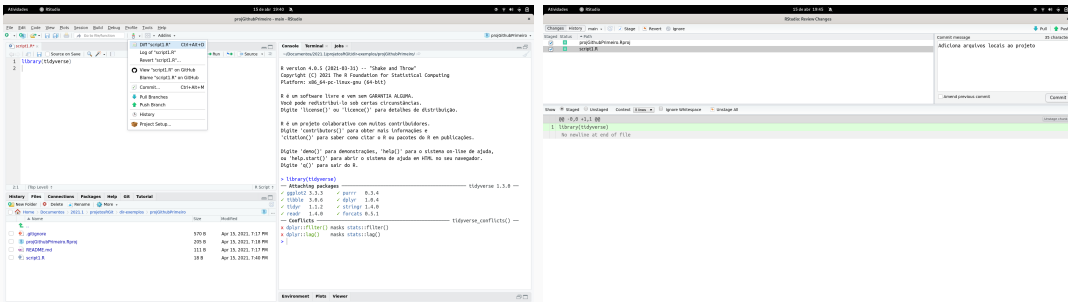


Figura 16: repositório local criando antes

RStudio/Git: Ferramentas gráficas

O RStudio possui algumas facilidades para lidar com Git e Github nos seus projetos.

- Alguns comandos estão disponíveis a um click



Lista 9 - Usando comandos git com RStudio

- Crie um novo projeto e adicione o repositório remoto no github
- Insira/altere arquivos localmente e sincronize as alterações usando a interface

- Pro Git book
- Documentação do Github
- Documentação do Gitlab
- Reproducible Research with R and R Studio
- Replicação, Replicação Gary King
- Open Science Framework
- BITSS
- Protocolo TIER
- Framework for Open and Reproducible Research Training