



Introducción a Git y GitHub con R

Joselyn Chávez

 [@josschavezf1](https://twitter.com/josschavezf1)

RLadies Bariloche

01 de junio de 2022

¿Qué es Git?

- Git es un software libre que funciona como un sistema de control de versiones.
- Permite guardar y compartir código.
- Permite regresar a versiones anteriores, revisar cambios y crear ramas experimentales para probar código sin dañar el código principal.



Instalación

Git se puede utilizar en todos los sistemas operativos.

Algunas opciones para instalar git en tu sistema:

- <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Instalaci%C3%B3n-de-Git>
- <https://git-scm.com/downloads>
- <https://github.com/git-guides/install-git>
- <https://gitforwindows.org/>

Algunas plataformas que utilizan git



<https://github.com/>




<https://gitlab.com>

Creando mi primer repositorio de 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 ^{*}

 josschavezf / mi_primer_repositorio ✓

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

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.

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: **None** ▾

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

License: **None** ▾

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

① You are creating a public repository in your personal account.

[Create repository](#)

Conectando GitHub con RStudio

Paso 1: Generar un token

Settings > Developer
settings > Personal access
tokens > Generate new
token.

Seleccionar: repo,
write:packages y admin:org



Conectando GitHub con RStudio

Paso 2: Guardar las credenciales en R

Copia el token

Ejecuta el comando:

```
gitcreds::gitcreds_set()
```

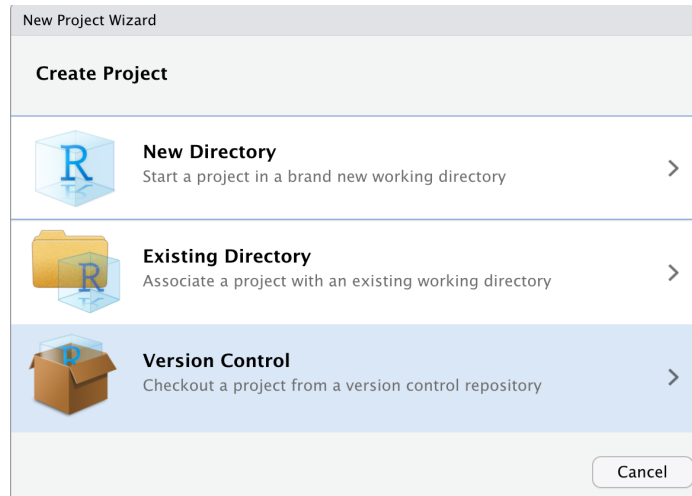
Pega el token, confirma tus credenciales y roles

```
gh::gh_whoami()
```



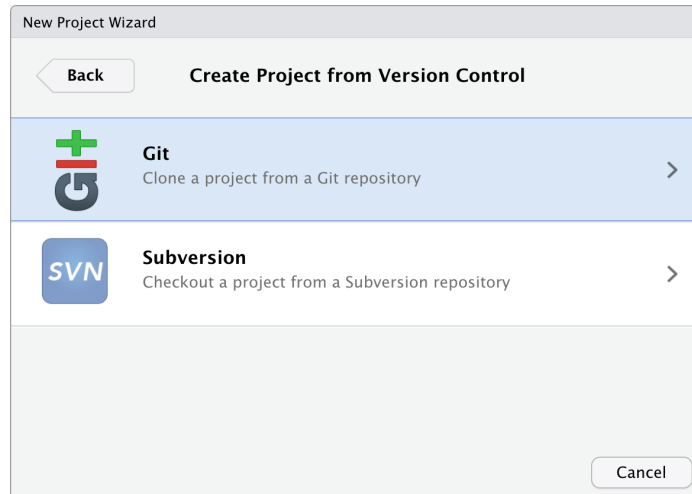
Clona el repositorio en tu computadora

File > New Project > Version control



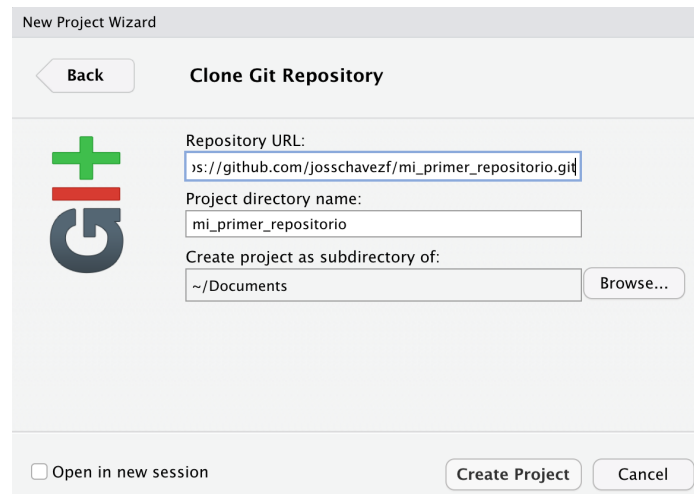
Clona el repositorio en tu computadora

Selecciona la opción Git



Clona el repositorio en tu computadora

Pega el link del repositorio y selecciona la ubicación que tendrá



The screenshot shows the 'New Project Wizard' dialog box in RStudio, specifically the 'Clone Git Repository' step. The dialog has a title bar 'New Project Wizard' and a 'Back' button. On the left is the Git logo. The main area contains three input fields: 'Repository URL:' with the value 'rs://github.com/josschavezf/mi_primer_repositorio.git', 'Project directory name:' with the value 'mi_primer_repositorio', and 'Create project as subdirectory of:' with the value '~/Documents'. There is a 'Browse...' button next to the last field. At the bottom, there is a checkbox 'Open in new session' which is unchecked, and two buttons: 'Create Project' and 'Cancel'.

New Project Wizard

Back

Clone Git Repository

Repository URL:
rs://github.com/josschavezf/mi_primer_repositorio.git

Project directory name:
mi_primer_repositorio

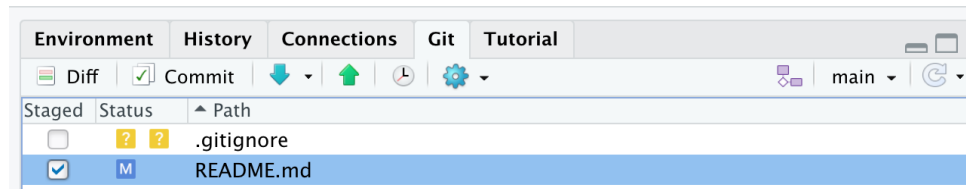
Create project as subdirectory of:
~/Documents Browse...

☐ Open in new session

Create Project Cancel

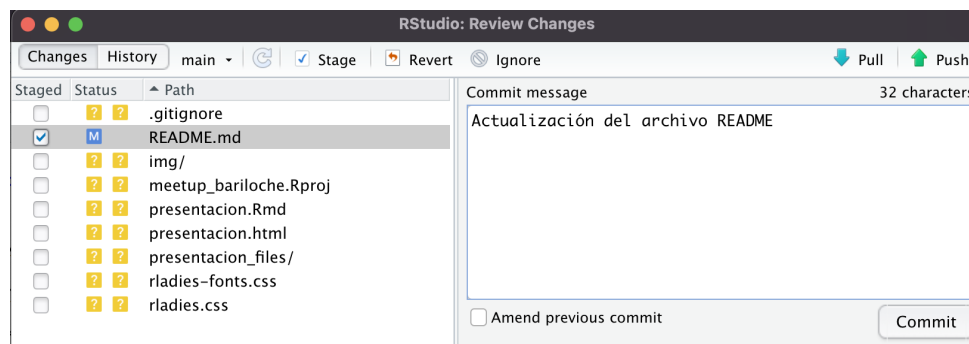
Creando cambios en el repositorio

- Abre el archivo README.md y agrega algunas líneas.
- Guarda los cambios.
- Ve a la pestaña Git y selecciona la casilla del archivo.



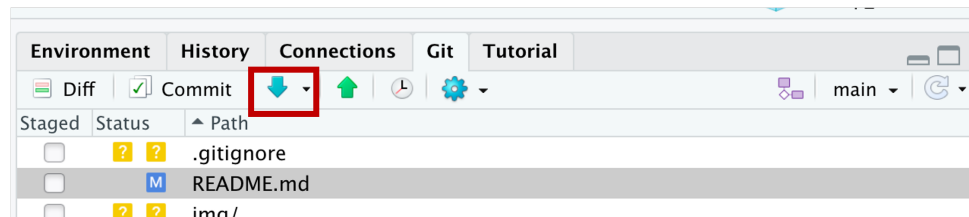
Subiendo cambios en el repositorio

- Selecciona la opción Commit.
- Escribe un mensaje de rastreo para los cambios generados.
- Selecciona el botón Commit y después Push.
- Verifica los cambios en el repositorio en línea.



Importa cambios desde el repositorio

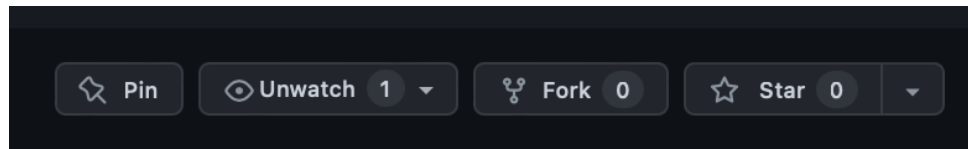
- Realiza algunos cambios en el archivo README.md desde la página de GitHub.
- Importa los cambios a la computadora usando el botón Pull.



Creando colaboraciones

Fork y Pull-request

- Ve al repositorio
https://github.com/josschavezf/repositorio_prueba
- Selecciona el botón fork
- Genera el fork en tu usuario de GitHub



- Clona el repositorio resultado del fork, debe lucir similar a: https://github.com/tu_usuario/repositorio_prueba
- Crea un nuevo script y llámalo con tu nombre

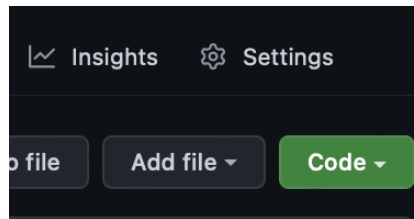
```
1 x <- 1:5
2
3 x[3:5]
4 |
```

- Guarda los cambios.
- Haz un commit y push
- Ve a tu repositorio en github.com dentro de tu usuario y selecciona Pull-request

Creando colaboraciones

Agregando colaboradores al repositorio

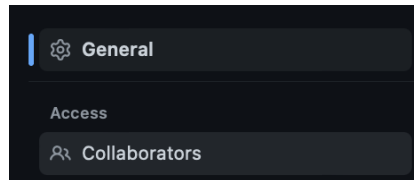
- Ve al primer repositorio que creaste, `mi_primer_repositorio`
- Ve a la sección settings del repositorio



Creando colaboraciones

Agregando colaboradores al repositorio

- Selecciona la sección de colaboradores en el panel izquierdo.



- Selecciona Add people.
- Escribe el nombre de usuario de GitHub de tu colaborador/a.
- Selecciona el rol que tendrá tu colaborador/a.

Creando colaboraciones

Agregando colaboradores al repositorio

- Tu colaborador/a puede clonar tu repositorio directamente, sin hacer fork ni pull-request.
- Pide a tu colaborador/a que clone tu repositorio en su computadora.
- Tu colaborador/a generará cambios en el repositorio clonado y subirá el cambios haciendo commit y push.
- Verifica que los cambios se ven reflejados en tu repositorio.

¡Gracias!



Ilustración de
Allison Horst