

---

# Tutoriales para instalación de RStudio y GIT y su integración

Introducción al procesamiento, análisis y visualización interactiva de datos abiertos en salud - UNTREF

---

[Instalar R y RStudio:](#)

[Integrar GIT con R y Rstudio](#)

- [1. Instalar Git](#)
- [2. Crear cuenta de GitHub](#)
- [3. Configure Git en RStudio:](#)
- [4. Configure Git:](#)
- [5. Crear un repositorio propio en GitHub](#)
- [6. Clonar un repositorio, o conectar un proyecto con un repositorio GitHub](#)
- [7. Agregar archivos a su repositorio y realizar un commit](#)

## Instalar R y RStudio:

Instalación de R: Permite comprender el lenguaje de programación R

Instalación de RStudio: Se instala un entorno de desarrollo integrado (IDE) que facilita el uso de R. Es el IDE más usado para el lenguaje R.

Para instalar R ingresa [aca](#) y sigue las instrucciones:

---

## 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 (2020-10-10, Bunny-Wunnies Freak Out) [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 [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.

## R for Windows

Subdirectories:

[base](#)

[contrib](#)

[old contrib](#)

[Rtools](#)

Binaries for base distribution. This is what you want to [install R for the first time](#).

Binaries of contributed CRAN packages (for R  $\geq$  2.13.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

Binaries of contributed CRAN packages for outdated versions of R (for R  $<$  2.13.x; managed by Uwe Ligges).

Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

R-4.0.3 for Windows (32/64 bit)

**Download R 4.0.3 for Windows** (85 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is [CRAN.MIRROR>/bin/windows/base/release.html](https://CRAN.MIRROR>/bin/windows/base/release.html).


---

Last change: 2020-10-10


Una vez instalado R, pueden instalar RStudio:

RStudio es un entorno integrado de desarrollo (IDE por sus siglas en inglés). Si bien, podemos utilizar R directamente desde documentos de texto plano, se recomienda instalar y usar un IDE, ya que es una manera más práctica para trabajar y desarrollar nuestros proyectos. Esto se debe a que nos facilitará tareas como administrar los archivos con los que trabajamos, emplear herramientas para escribir y revisar nuestro código, efectuar capacidades gráficas, así como fluidez para trabajar en el entorno local de nuestro ordenador.

Ingresa al sitio oficial de [RStudio](https://posit.co/).



[PRODUCTS](#) [SOLUTIONS](#) [LEARN & SUPPORT](#) [EXPLORE MORE](#) [PRICING](#)



## 1: Install R

RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

[DOWNLOAD AND INSTALL R](#)

## 2: Install RStudio

[DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS](#)

Size: 212.77 MB | [SHA-256: A8325AD5](#) | Version: 2023.06.1+524 | Released: 2023-07-07

## Integrar GIT con R y Rstudio

R en combinación con el sistema de control de versiones distribuidas Git ofrece una configuración conveniente para hacer que su proyecto de investigación sea reproducible. Git le permite rastrear y compartir su código y análisis de manera eficiente.

Algunas razones para utilizar GIT y GitHub son:

- Facilita el compartir sus proyectos de forma sencilla una vez que estén configurados correctamente.
- Favorece la colaboración, permitiendo a otras personas contribuir a su proyecto y viceversa. Además, pueden reportar errores o sugerir nuevas incorporaciones al proyecto.
- Le permite volver a versiones anteriores en caso de encontrar errores o eliminar algo accidentalmente.
- Permite visualizar los cambios entre diferentes versiones de su código, análisis o texto escrito.

**GitHub** es un servicio web fácil de usar que le permite almacenar el repositorio de su proyecto de forma remota. Otras alternativas son GitLab y Bitbucket.

RStudio integra soporte para Git, por lo que utilizaremos la combinación ampliamente utilizada R + Git + RStudio para el control de versiones y desarrollo colaborativo de proyectos.

### 1. Instalar Git

Desde el siguiente enlace: <https://git-scm.com/downloads>

### 2. Crear cuenta de GitHub

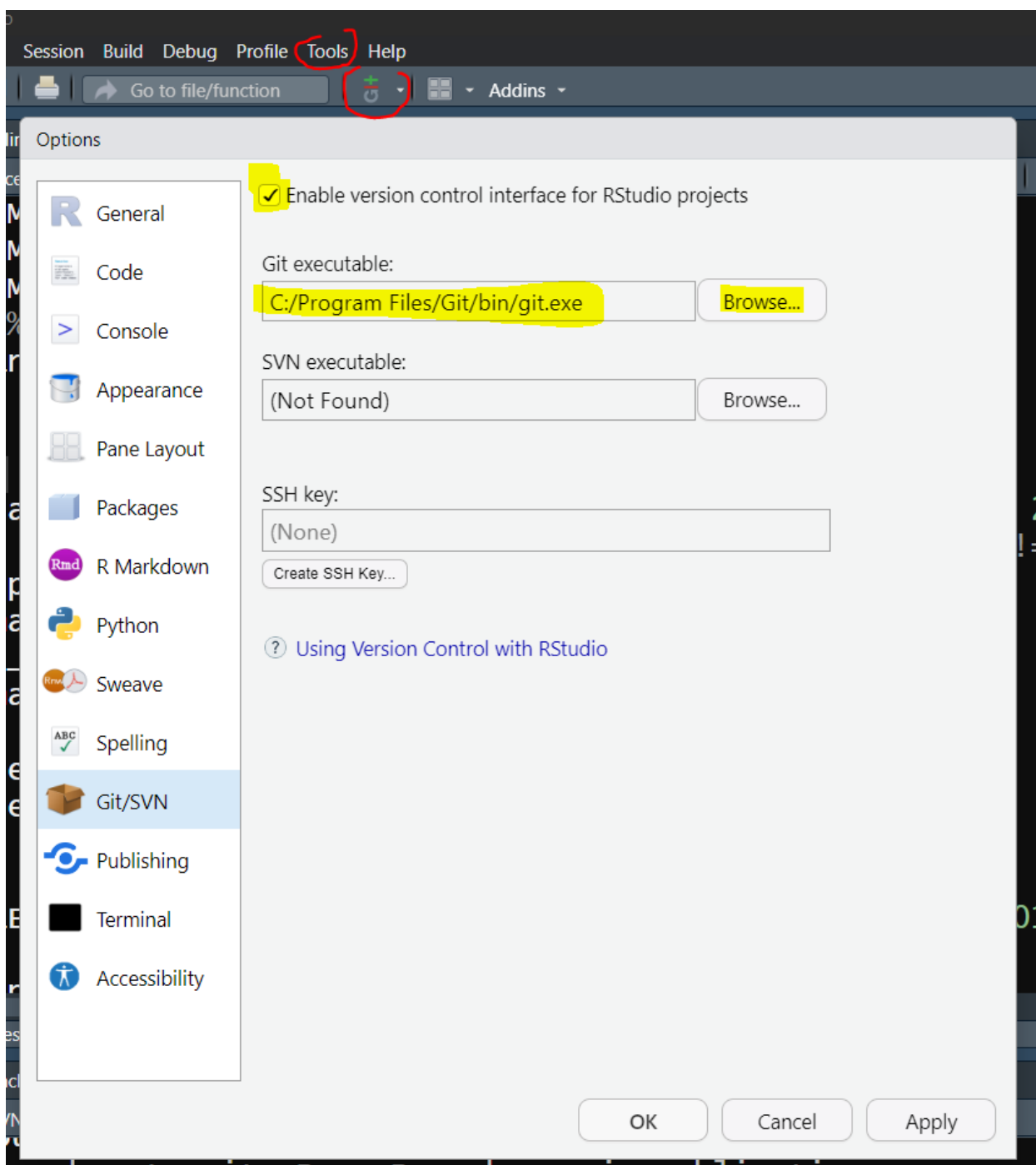
Crear su cuenta desde el siguiente enlace <https://github.com/>

### 3. Configure Git en RStudio:

Indique a RStudio dónde encontrar la instalación de Git. Para ello sigue los siguientes pasos:

1. Abra RStudio y vaya a Herramientas > Opciones globales... haga clic en Git/SVN
2. Marque Habilitar interfaz de control de versiones para proyectos de RStudio
3. Establezca la ruta al ejecutable de Git que acaba de instalar. Puede ayudarse del boton "Browse.." para llegar a la ruta. En la imagen un ejemplo de ruta.

4. Reinicie RStudio. Si funcionó, encontrará el icono de Git en la barra de herramientas superior, como se muestra a continuación.



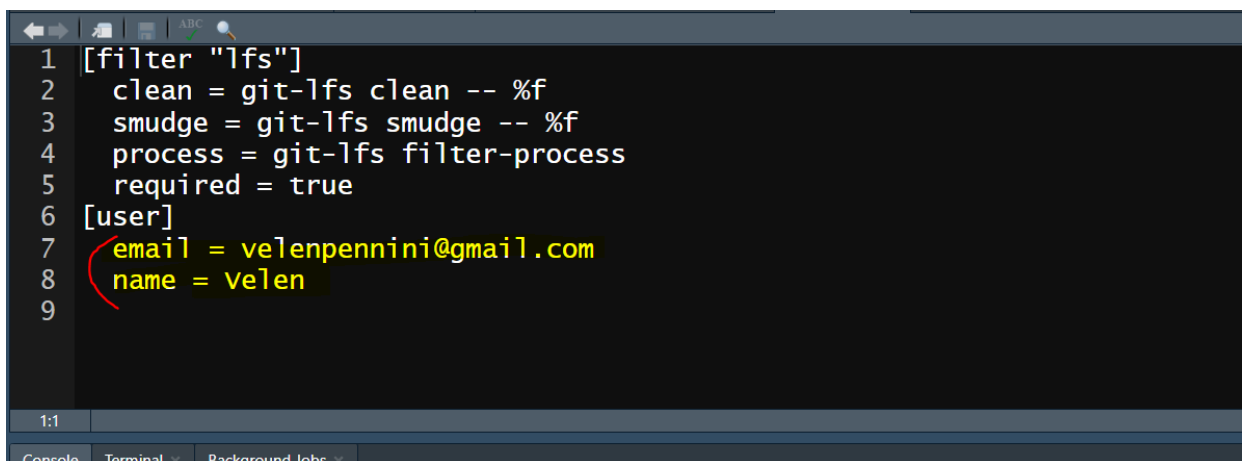
#### 4. Configure Git:

En la consola de RStudio escribir

```
library(usethis)
```

```
edit_git_config()
```

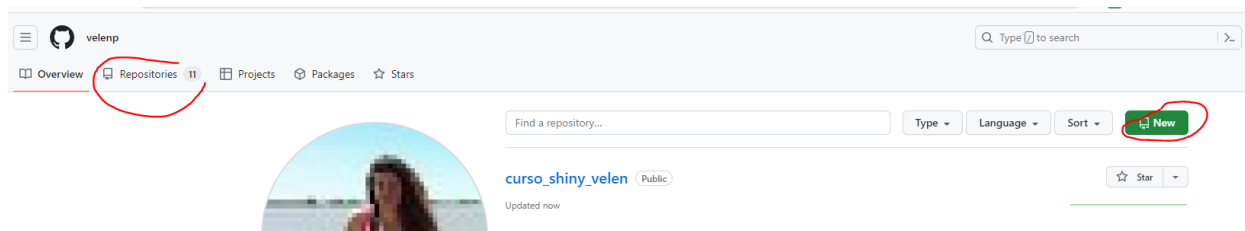
Luego verán que se crea una ventana llamada “.gitconfig” y allí deberán completar los campos *name* y *email* tal como se ve en la figura y guardar.



```
1 [filter "lfs"]
2   clean = git-lfs clean -- %f
3   smudge = git-lfs smudge -- %f
4   process = git-lfs filter-process
5   required = true
6 [user]
7   email = velenpennini@gmail.com
8   name = Velen
9
```

## 5. Crear un repositorio propio en GitHub


- Ingresar a la pagina de GitHub y crear un repositorio nuevo que se llame “curso\_shiny\_sunombre” agregandole su nombre al final. Al hacer clic en “new” se abrirá una pantalla como la que está a continuación.



## Create a new repository

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


Required fields are marked with an asterisk (\*).


Owner \*  velenp / Repository name \*

✓ curso\_shiny\_ven is available.

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

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:

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

Add .gitignore

.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **None**

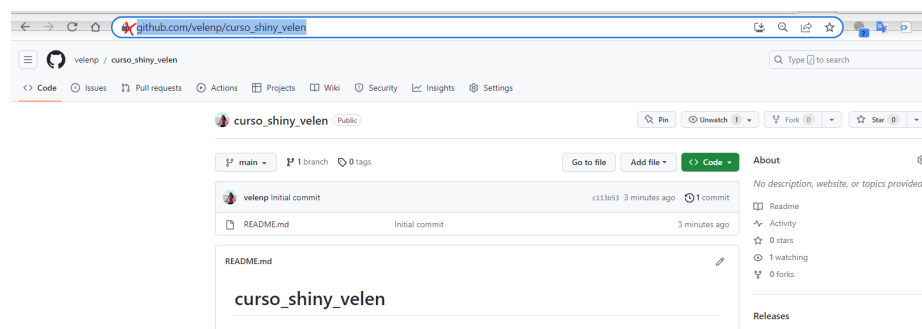
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

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

 You are creating a public repository in your personal account.

Create repository

- b. Ahora, increse al repositorio y copie el enlace, tal como se muestra en la captura.



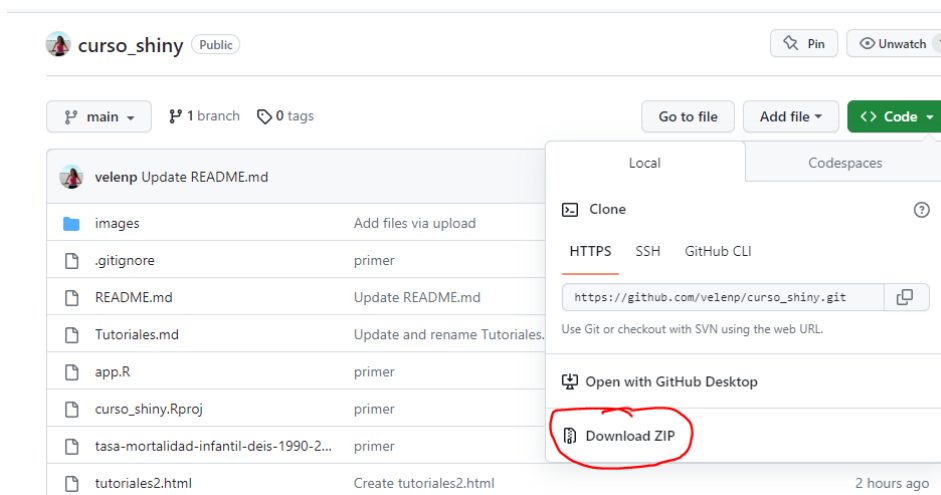
## 6. Clonar un repositorio, o conectar un proyecto con un repositorio GitHub

- a. Ir al repositorio que desea clonar. En este caso, el del curso es:

[https://github.com/velenp/curso\\_shiny](https://github.com/velenp/curso_shiny)

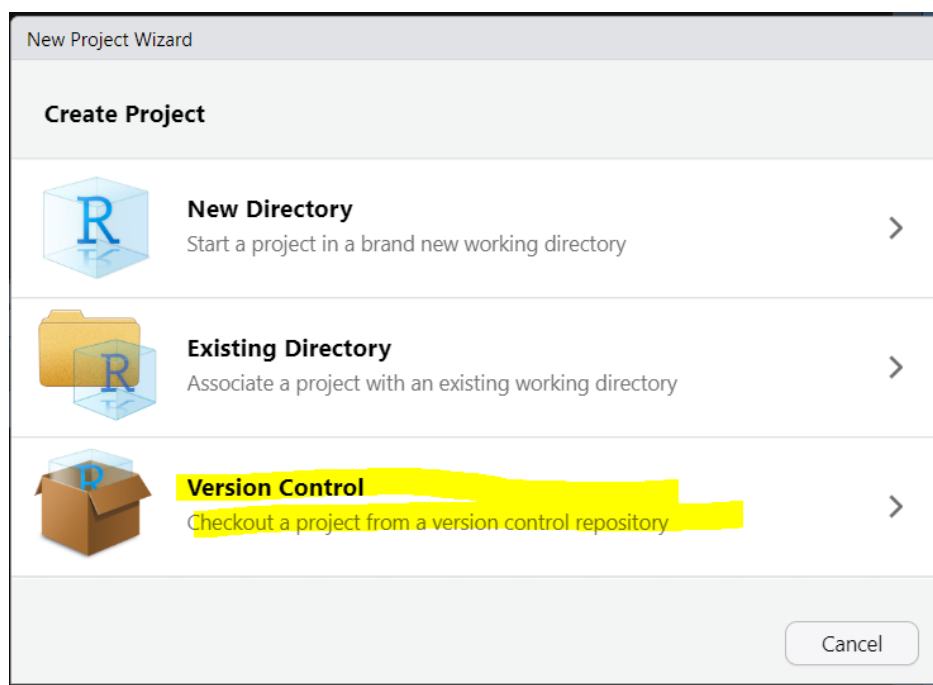
Haga clic en el boton verde que dice “code” y se desplegará una ventana, donde puede copiar el enlace del repositorio, o también puede descargar el .ZIP. En este caso vamos a descargar el .ZIP. Si usted desea conectarse para trabajar colaborativamente con ese repositorio puede copiar el enlace y clonarlo desde RStudio. Hay que tener en cuenta que para que esto sea posible el repositorio debe estar compartido con su usuario, de lo contrario no podrá trabajar colaborativamente.

En este caso, lo que vamos a hacer es clonar el repositorio vacío que ustedes crearon y a ese repositorio del que ustedes son propietarios subirles los materiales que se descargaron desde aquí en un .ZIP.

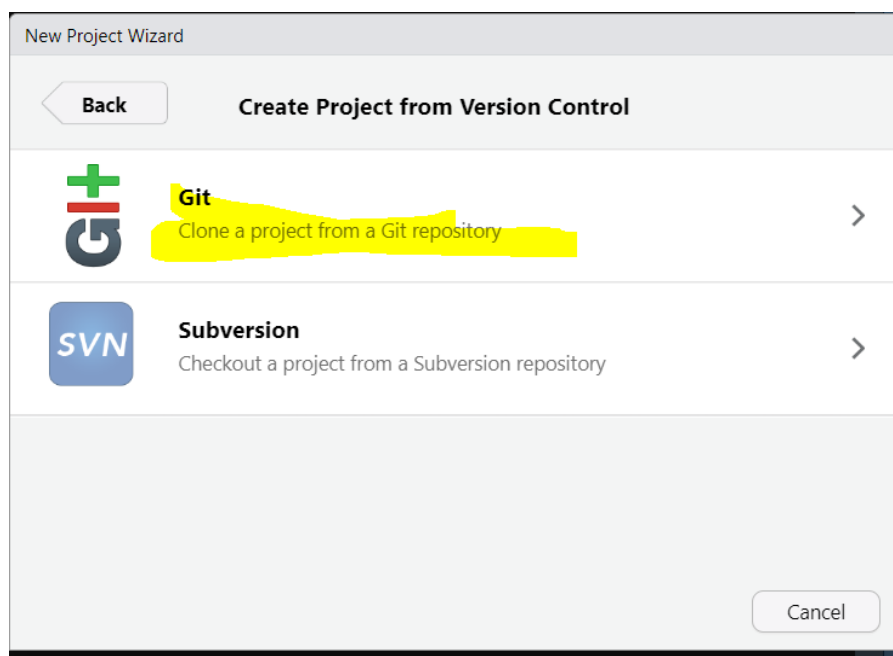


- b. En RStudio, en el margen superior derecho, ir a “nuevo proyecto” o “New Project”.
- c. En la pantalla que se abre seleccione la opción “Version Control” tal como se ve en la captura a continuación.

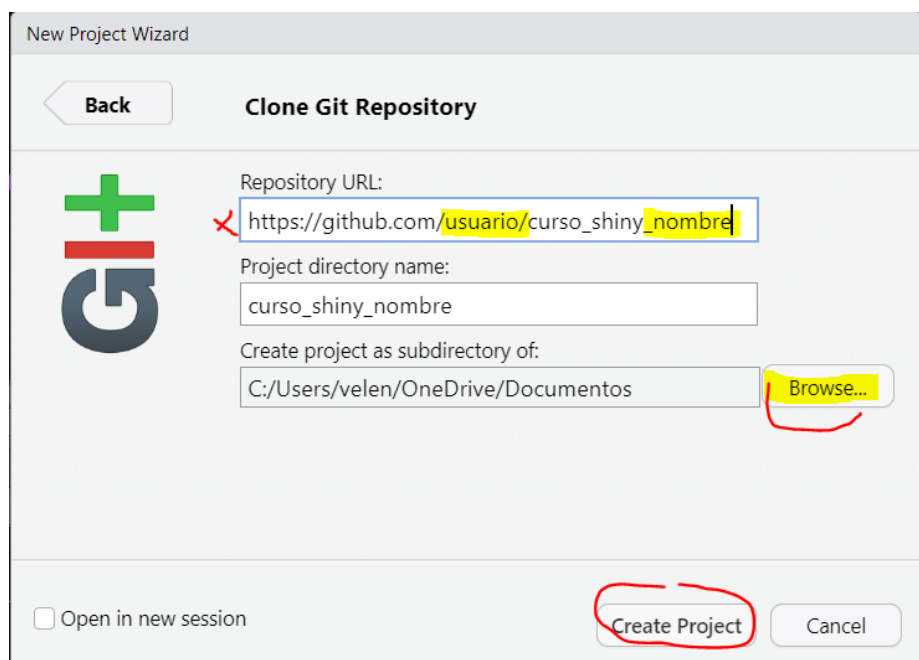




Seleccione la opción GIT

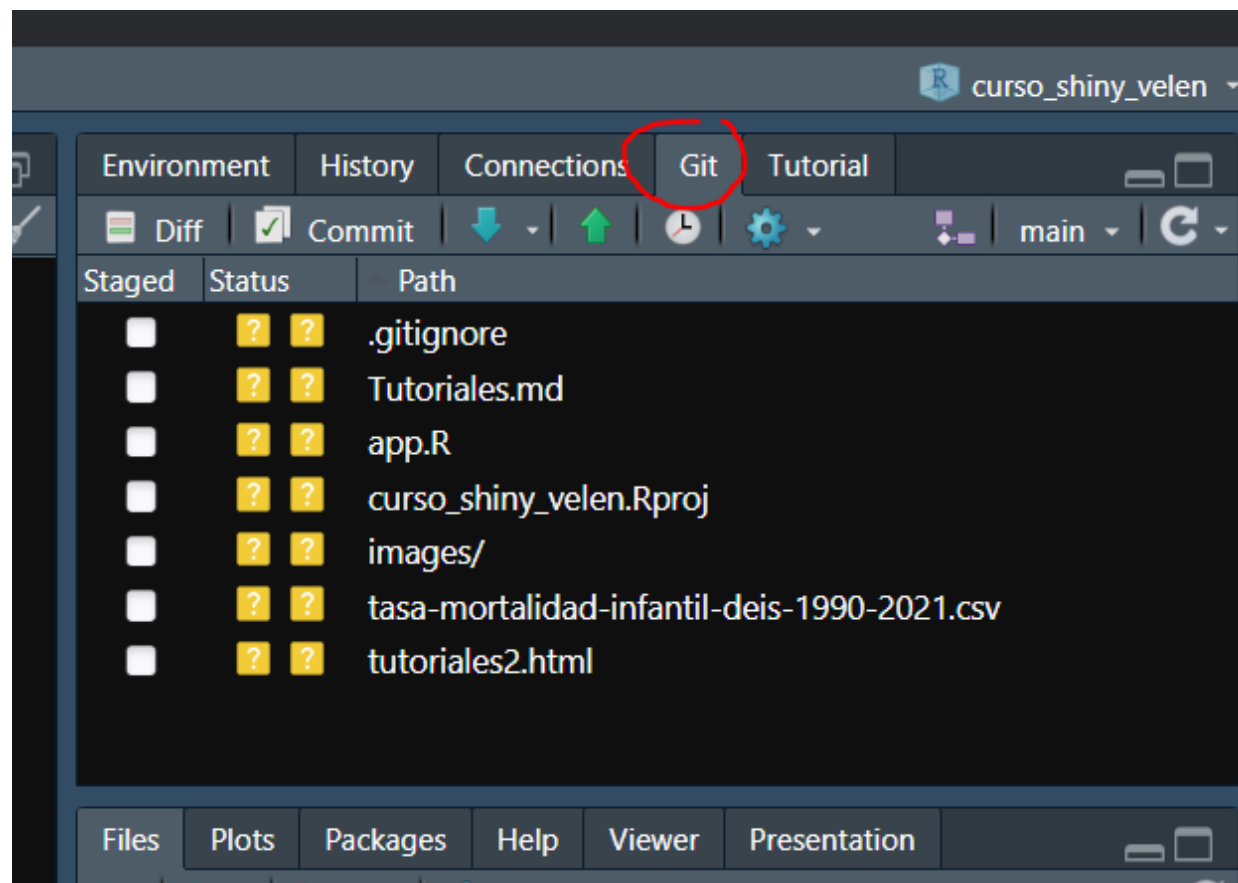


Ingresa la URL del repositorio que tiene su nombre, que crearon en el paso 5, seleccione la carpeta donde lo quiere alojar y ponga crear proyecto.

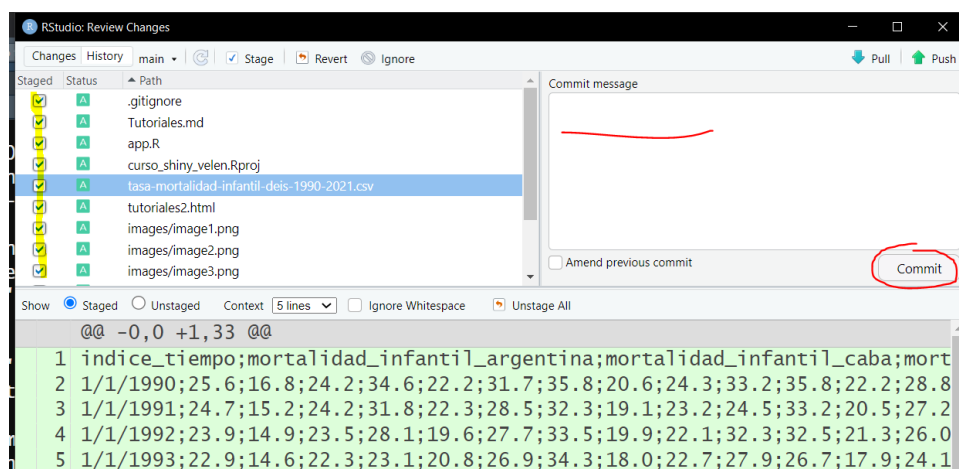


## 7. Agregar archivos a su repositorio y realizar un commit

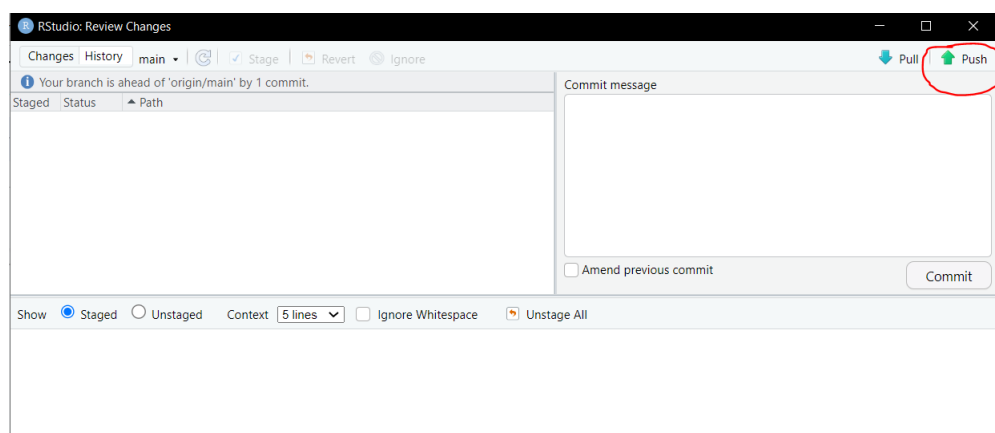
- En su computadora, ahora encontrará una carpeta llamada "curso\_shiny\_nombre" con el contenido del curso. En esta carpeta deberán pegar el contenido de la carpeta del archivo.ZIP. Importante: no reemplazar archivos y eliminar el archivo "curso\_shiny.Rproj" de modo que quede un solo archivo con extensión .Rproj en la carpeta, y este tenga su nombre.
- Si vuelve a RStudio, verán en la ventana GIT, que se encuentra como solapa en el mismo espacio que el Environment, todos estos archivos nuevos agregados.




- c. Estos archivos se encuentran guardados localmente. Si quieren subirlos a su repositorio de GitHub deberá hacer un "commit" y luego "Push". Veamos el paso a paso:
- d. Hacer clic en Commit, y verán que se despliega una ventana como la presentada a continuación. Allí, tildar todos los archivos tal como se ve en la captura. Donde dice "commit message" poner un mensaje que identifique a este commit, como por ejemplo "primer commit". y por ultimo hacer clic en "commit" que se muestra en la figura señalado en rojo.



- e. Aparecerá una ventana negra con el detalle de la tarea realizada. Cerra esa ventana y hacer clic en el botón de “push” con una flecha verde hacia arriba, tal como se muestra en la captura.



- f. Ahora, si ingresan al su repositorio desde la pagina de GitHub verán que contiene los archivos que subieron. Cada vez que modifiquen alguno de estos archivos o agreguen un archivo nuevo deberán repetir el proceso de “Commit” y luego “Push” para actualizar su repositorio de GitHub. En texto que pongan en “commit message” será una referencia al momento de revisar el historial de cambios que tuvo el repositorio.
- g. En caso que estén trabajando con un repositorio de forma colaborativa, con otras personas, es necesario, que antes de realizar cada “commit” y “push” presionen el botón de la flecha celeste hacia abajo que dice “Pull”. De este modo podrán incorporar los cambios que realizó otro colaborador en su proyecto. Es importante realizar el “Pull” antes que el “Push” para evitar conflictos entre versiones. Si varios colaboradores están trabajando en el mismo proyecto y realizan cambios en la misma parte del código, un “pull



request" permite resolver posibles conflictos antes de que se mezclen los cambios. Esto evita que los cambios se fusionen con conflictos no resueltos, lo que podría llevar a errores en el código.