

Pasos para instalar Git y configurarlo en RStudio:

### 1. Instalar Git:

- En Windows:
  - Descarga el instalador de Git desde el sitio web oficial: <https://git-scm.com/downloads>
  - Ejecuta el instalador descargado y sigue las instrucciones en pantalla para completar la instalación. Asegúrate de seleccionar "Git Bash Here" durante la instalación para tener acceso a la línea de comandos de Git.
- En macOS:
  - Git generalmente ya está instalado en macOS. Sin embargo, si deseas una versión más reciente, puedes instalar Git utilizando Homebrew o descargarlo desde el sitio web oficial mencionado en el punto anterior.
  - Para instalar Git con Homebrew, abre una terminal y ejecuta el siguiente comando:

- ☐
- brew install git

### ☐ En Linux (basado en Debian/Ubuntu):

- Abre una terminal y ejecuta el siguiente comando:

sql

1.

- - sudo apt-get update
  - sudo apt-get install git
  -

### 2. Configurar Git en RStudio:

- Abre RStudio en tu sistema.
- Ve al menú "Tools" (Herramientas) y selecciona "Global Options" (Opciones globales).
- En la ventana de "Options" (Opciones), selecciona "Git/SVN" en el panel izquierdo.
- Haz clic en el botón "Browse" (Examinar) junto a la opción "Git executable" (Ejecutable de Git).
- Navega hasta la ubicación donde se instaló Git en tu sistema y selecciona el archivo ejecutable de Git (por ejemplo, git.exe en Windows o git en macOS/Linux).
- Haz clic en "OK" para guardar la configuración.

Ahora, Git está instalado y configurado en RStudio. Puedes usar las funciones y comandos de Git en RStudio para administrar tus repositorios y versiones de código.

Para validar si ya tenemos instalado Git en Rstudio ir a la pestaña de la terminal en Rstudio y escribir el comando "git -v". Debe aparecer la versión de Git "git versión 2.41.0.windows.2"

Para configurar Git en la terminal, escribir nuestro correo y usuario de GitHub con los siguientes comandos

Ejemplo:

```
git config --global user.email "nombredeusuario@correo.com"
```

```
git config --global user.name "escriba su nombre de usuario sin espacios"
```

Nota: no escribas las comillas en el comando

Para establecer directorio interno escribimos el comando "cd". Ejemplo: cd  
C:/Users/erdav/Documents/Rstudio

Para ver nuestros archivos de nuestro directorio ejecutamos el comando "ls"

Creamos el repositorio dentro de esa carpeta con el comando: git init

Establecemos como rama principal con el comando: git branch -M main

Establecemos el link donde queremos alojar nuestro repositorio: git remote add origin  
<https://github.com/erdavg/CursoR2023.git>

Para corroborar en que repositorio nos encontramos en el perfil de github: git remote -v

Con este comando vemos el estado de nuestros archivos: git status

Ejecutamos este comando para trackear todos nuestros archivos que no están actualizados, en este caso los que no están actualizados aparecerán en color rojo : git add .

Volvemos a utilizar nuestro comando git status para asegurarnos de que nuestros archivos fueron actualizados, debería de ser vistos en color verde

Con este comando ponemos algún comentario de los cambios que se actualizo dentro de esos archivos. Ejemplo: git commit -m "se agregaron gráficas al proyecto"

Usamos este comando si es la primera vez que se sube algo al repositorio: git push --set-upstream origin main

Usamos este comando para actualizar o verificar nuestro repositorio: git push

Verificar en nuestro perfil de GitHub que nuestros archivos se subieron con éxito

Para ver el historial de versiones de nuestros commits: git log

Si queremos regresar a una versión pasada copiamos el identificador del commit y el comando git checkout. Ejemplo:

```
Git checkout 1982626f880e35a534f1e6cebd49d93b49e95e7a
```

Si llamamos de nuevo al comando git log solo nos aparecerá HEAD y no MASTER o MAIN. Eso significa que dejamos de estar en la rama principal.

Para poder visualizar nuestras ramas llamamos a: git Branch

Si queremos volver a master usamos: git switch master

