

# Guía básica para el uso de Git y GitHub

Gobierno de Datos

## 0.1. Introducción

Con el objetivo de facilitar el trabajo transparente y colaborativo entre los equipos de Gobierno de Datos, se ha generado esta Guía para el uso de herramientas que permitan conservar un registro del control de versiones de proyectos en que participan varias personas. La estructura de este documento es la siguiente: primero se muestran instrucciones sobre la instalación de Git y posteriormente se explica cómo configurar e iniciar un proyecto desde R studio.

### 0.1.1. Github



**Figura 1:** GitHub

GitHub se nombra como la “plataforma de desarrollo de software líder en el mundo”, pues permite publicar, compartir y socializar (como una Red Social Digital) el código de programación. Además, permite registrar y ver cada cambio en el código y las carpetas que se ha hecho, permitiendo rastrear el progreso y, si es necesario, revertirlo (2019).

### 0.1.2. Crear cuenta y repositorios GitHub

Acceder al siguiente link para crear cuenta de GitHub [Crear Cuenta](#).

1. Una vez creada la cuenta, en la parte superior derecha de la página de Git Hub se observará un símbolo ‘+’.
2. Debe abrir esa pestaña y de clic en “New Repository”.
3. Ahora aparecerán las opciones de configuración donde tendrá la opción de agregar un documento README.md.
4. Listo, una vez seleccionadas las preferencias de configuración el nuevo repositorio se habrá creado.

Por otro lado, si no es necesario hacer un repositorio nuevo, se solicitará el enlace del repositorio en que quiera trabaja, el cual se podrá obtener al dar clic en el botón verde ‘Code’ para, posteriormente, clonar el repositorio. Guárdese, se empleará en otra etapa.



Figura 2: git

### 0.1.3. Instalación de Git en Linux y Windows

Para descargar el archivo de instalación de Git acceder al siguiente link [Downloads](#).

Sobre la instalación, las opciones predeterminadas permitirán el funcionamiento adecuado de Git en el ordenador. Como recomendación, se podrá crear una carpeta “repositorio” en la siguiente ubicación:

C : / Usuarios / nombre de usuario / Documents / repositorio\_nuevo

En esta ubicación se generará el repositorio que podrá usarse en el futuro. Una vez instalado Git, se deben de seguir los siguientes pasos para copiar un repositorio. Como amigable recordatorio, la forma de correr una línea de código generalmente es pulsando **Ctrl + Enter**. Y el primer paso es Abrir Git Bash y escribir:

- cd Documents
- cd nuevo\_repositorio

Lo anterior agrega la carpeta *nuevo\_repositorio* a git, reconociéndola como repositorio. Posteriormente, para clonar el repositorio de GitHub:

- git clone URL

Donde **URL** es el enlace para clonar desde la página de GitHub.

**0.1.3.1. Autenticación** Para vincular un repositorio de GitHub a RStudio desde Git, será necesario el enlace del repositorio en que quieras trabajar. Se explicará cómo usar este enlace en una etapa posterior.

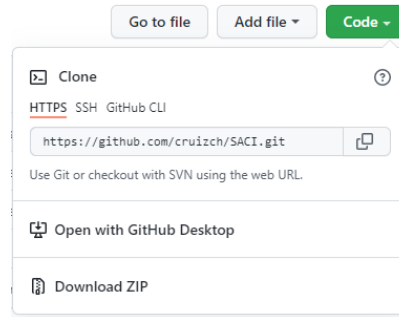
A este procedimiento se le conoce como autenticación. Es importante mencionar que, existen dos opciones para vincular un repositorio. Es posible usar el formato de HTTPS o SSH los cuales tienen las siguientes características.

Se sugiere que el usuario emplee el protocolo HTTPS, por su eficiencia y facilidad.

- **HTTPS:** Si se elige el formato HTTPS, esto le permitirá usar un administrador de credenciales para almacenar en memoria caché las credenciales de GitHub en Git.  
Para seguir este protocolo se sugiere seguir lo siguientes pasos, retomados por la guía oficial de GitHub:
  1. Instalar GitHub CLI
  2. En la terminal de GitBash, escribir gh auth login y seguir las instrucciones.
- **SSH:** Al elegir la opción SSH, será necesario crear claves SSH en cada ordenador que use para su proyecto, ya sea para enviar o extraer información del repositorio en GitHub. Para establecer la configuración SSH o cambiar de HTTPS a SSH consultar el siguiente [enlace](#)

Para obtener estos enlaces debes dirigirte a tu repositorio en GitHub y dar clic al botón verde “Code”.

Más información puede consultarse el siguiente: [enlaceSSH](#)



**Figura 3:** Rmarkdown

## 0.2. Trabajar con Git desde Git Bash

Al abrir Git Bash, seleccionar el repositorio o carpeta donde se encuentren los archivos que se desean actualizar:

- `cd / Documents/nuevo_repositorio`
- `git add nombre_archivo`

## 0.3. Actualización o descarga de repositorio local

Para estar al tanto de los cambios que se han realizado en el repositorio de Github, se debe hacer una actualización de los cambios realizados en la nube. Por otro lado, cuando se quiere subir cambios a una rama, antes de subir los cambios (*push*) se debe actualizar (*pull*), ya que en caso de que haya cambios no registrados en el repositorio local podría haber un error y no ejecutar la función ***push***.

Procedimiento para actualizar:

```
git pull --verbose
```

### 0.3.1. Publicar cambios en el repositorio remoto GitHub

Para agregar las modificaciones realizadas a todos los archivos contenidos en el repositorio local se debe seguir el siguiente procedimiento. Sin embargo, se recuerda que antes de publicar cambios, es necesario trabajar sobre el archivo. Esto se puede ver en la sección de [RStudio1](#).

```
git add
```

```
git commit -m "Agregar un Mensaje de qué es lo que se realizó" git push
```

### 0.3.2. Publicar cambios en un solo archivo

Si se requiere publicar los cambios realizados en un solo archivo del repositorio el procedimiento es el siguiente:

```
git add archivo (Nombre del archivo ejemplo acb.txt)
```

```
git commit -m "Mensaje"
```

```
git push
```

### 0.3.3. Comandos Git

Para facilitar el uso de Git y resolver posibles dificultades con el uso de funciones, se agrega una sección con los comandos Git y sus funciones más utilizadas.

Comando	Función
<code>git clone URL</code>	Descargar un repositorio existente
<code>git branch &lt;branch name&gt;</code>	Crear nuevo Branch
<code>git branch --list</code>	Visualizar Branch existentes
<code>git branch -d &lt;branch-name&gt;</code>	Eliminar Branch
<code>git checkout &lt;name of your branch&gt;</code>	Para cambiar de una rama a otra (1)
<code>cd carpeta/</code>	Eliminar un repositorio local
<code>rm -rf .git</code>	
<code>git rm -r --cached directorio</code>	Eliminar repositorio y contenido
<code>git rm archivo</code>	Eliminar un archivo en particular

(1) Para usar este comando, previamente se debe verificar o guardar los cambios en la Rama actual. La rama que se desea verificar debe estar en su repositorio local.

En caso de querer profundizar en el manejo de esta herramienta y sus comandos avanzados, se sugiere descargar la guía oficial de Git a través del siguiente link [Downloads](#).

## 1. Vincular Git con Rstudio



Figura 4: RStudio

### 1.1. Instalar R y R Studio

Antes de tener RStudio, se debe de instalar previamente R. Esto se puede consultar aquí [Descargar\\_R](#).

Sobre las opciones de configuración, las que están predeterminadas permitirán el funcionamiento adecuado. Además, para conocer qué versión de R descargar, sea de 32 o de 64 bits, puede verificar siguiendo estos pasos:

1. Ingrese a **Configuración** de su ordenador
2. Seleccione **Sistema**
3. Pinche en **Acerca de...**
4. Visualice la opción de **Tipo de Sistema**. Ahí se identifica si es de 32 o 64 bits.

Una vez que se tiene R, es posible descargar RStudio desde aquí [Descargar\\_RStudio](#)

## 1.2. Preparar RStudio



**Figura 5:** funny

Para poder vincular GitHub con RStudio es necesario hacer un procedimiento.

0. Abrir Rstudio.
1. Ingresar la siguientes líneas de código dentro del ***R Script*** (es decir, en la sección superior izquierda)

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

2. Escribir su nombre de usuario y correo que usó para generar la cuenta en GitHub en la siguiente línea de código.

```
usethis::use_git_config(user.name = "Tu_nombre_Usuario_GitHub", user.email = "tucorreo@correo.com")
```

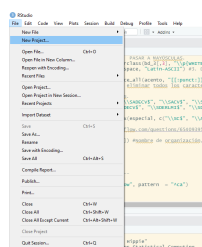
Recuerde reemplazar por los datos correctos del usuario.

Se sugiere que en este punto se tenga una cuenta de GitHub y un repositorio. Si no es así, puede ir a la sección sobre *Crear cuenta y repositorio en GitHub*, de este documento.

### 1.3. Uso de RStudio para clonar

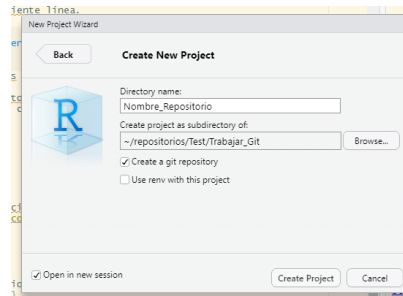
Para Vincular RStudio al proyecto en GitHub, es necesario seguir las siguientes instrucciones. Si se desea conocer más sobre la vinculación de Git y RStudio, se sugiere hacer click en la versión [extendida](#)

1. Hacer clic en "*Files*", luego "*New Project*".



**Figura 6:** Files

2. Dar clic en “*New Directory*”, posteriormente se verá un listado de opciones, seleccione “*New Project*”. Se abrirá una imagen relativa a crear un Nuevo proyecto.



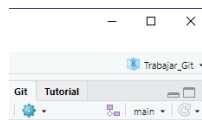
**Figura 7:** Nuevo proyecto

3. Una vez en la pestaña que se muestra, se debe escribir el nombre del Directorio, haga match en “*Create a git repository*” y se sugiere “*Open in new session*”.
4. Para terminar, se cierra el proyecto desde *File -> Close Project* antes de apagar RStudio a fin de evitar errores al recargar la interfaz en otra ocasión.

## 1.4. Trabajar en RStudio desde una rama local

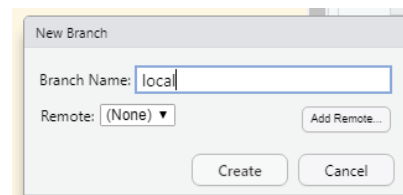
Para evitar problema al trabajar colaborativamente, tales como actualizaciones mientras se está trabajando sobre un repositorio, o dificultades al unir dos nuevas actualizaciones, se sugiere trabajar desde una **rama local**. También se sugiere seguir estos pasos, una vez que se sienta cómoda o cómodo trabajando en el entorno de Git y de RStudio.

1. **Importante** Se pinche en el símbolo morado para crear una rama local en la cual se trabajará.



**Figura 8:** 1

La cual se configura únicamente para trabajar de manera local como se muestra en la siguiente imagen.



**Figura 9:** 2

Este paso se realiza para evitar problemas cuando haya cambios en la rama principal main . De preferencia, se hace esta acción cada vez que inicie un nuevo proyecto. Para evitar errores más adelante, el nombre de la rama no debe contener símbolos especiales, ni puntos, ni acentos ni espacios: se sugiere usar un nombre del tipo *rama\_local* o bien *local*.

2. Se trabaja normalmente, y se realizan todas las acciones y actualizaciones dentro de esta rama local.

3. Se pincha Commit dentro de la pestaña *Git* y escribe los comentarios necesarios para explicar nuestro trabajo realizado en la ventana que se abre. Asegúrese de seleccionar solo los archivos que se deseen actualizar.
4. Junto al nombre de la rama local `rama_local` en la pestaña *Git* aparece un triángulo que apunta hacia abajo, se pinche y se cambia a la rama principal `main` . Esto abre un mensaje que, si no hay errores, indicará que ha cambiado a la rama principal.
5. Nuevamente se realiza pull porque podría haber actualizaciones realizadas por otro colaborador.
6. En caso de que haya actualizaciones después de *descargar* la última versión del repositorio, nuevamente se hace Commit . Note que ya se encuentra en la rama principal `main` del proyecto
7. Se abre una consola a partir del menú Tools -> Shell .
8. En la consola se escribe el siguiente comando: `git merge rama_local` donde `rama_local` es el nombre que se asignó a la rama local en la cual se trabajó. Esto se hace para *combinar* el trabajo local con la rama principal `main` antes de subir las actualizaciones al repositorio. Se cierra la consola.
9. A continuación, se hace push del trabajo. Esto se logra pinchando la flecha verde que apunta hacia arriba. Esto actualiza el repositorio.
10. Se elimina la rama local al escribir en la consola el siguiente comando: `git branch -d rama_local` . Una vez hecho esto, se cierra la consola.
11. Para concluir, se cierra el proyecto desde File -> Close Project antes de cerrar RStudio para evitar errores al cargar la interfaz en una próxima ocasión.
12. Cuando se reabra el proyecto una próxima vez, se puede comenzar desde acceder a una **rama local**. Disfrute :Happy\_face:

## 1.5. Sugerencias para el uso de R Markdown



Figura 10: Rmarkdown

Para tener mayor documentación conservamos las sugerencias de RStudio sobre RMarkdown: <http://rmarkdown.rstudio.com>.

## Referencias

- [1] Lapuerta, Madé, *GitHub For Dummies*, Medium (2019). Obtenido el 7 de enero de 2022 desde <https://madebymade.medium.com/github-for-dummies-96f753f96a59>