

Primeros pasos con Git

Miguel Equihua

Xalapa, Ver., 3 de julio, 2024



Figura 1: Fuente: *Final.doc* en **Piled Higher and Deeper** por Jorge Cham, <http://www.phdcomics.com>

¿Qué es Git?

Es una aplicación diseñada por el iniciador del desarrollo de Linux [Linus Torvalds](#). [Git](#) es un sistema eficiente confiable y distribuido de control de versiones. El control de versiones es simplemente el seguimiento y registro de los cambios que va teniendo un documento a lo largo del tiempo. El concepto *distribuido* se refiere a que el registro local que tengas en tu máquina o para el caso en cualquier número de máquinas, es un registro completo, **clonado** del proyecto. Estos repositorios locales plenamente funcionales permiten trabajar aún cuando no tengas acceso a Internet. Los autores realizan y registran su trabajo localmente y, cuando lo encuentren conveniente, sincronizan la copia local del repositorio con la del servidor. En la actualidad *Git* se ha convertido en el estándar mundial *de facto* para el control de versiones.

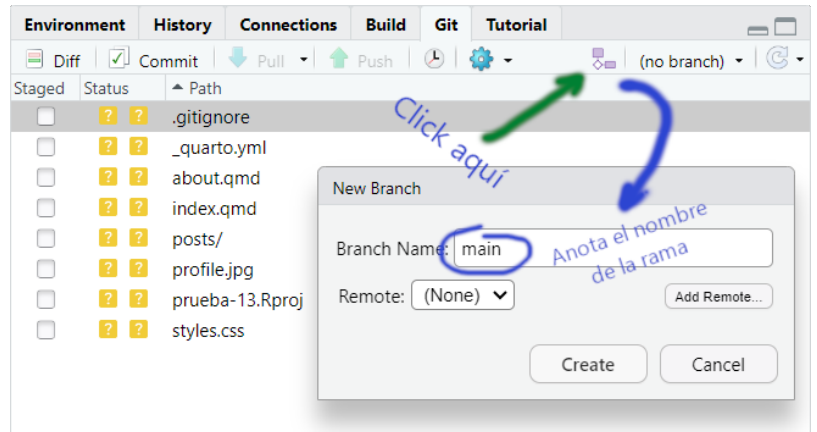
Una forma muy sencilla de instalar *Git*, incluyendo un auxiliar gráfico, es recurrir a [Github desktop](#). Otra forma es recurrir, si tu sistema operativo es Windows, a [Git for windows](#), que no incluye nada más que la plataforma básica de control de versiones. En todo caso, el prerequisite es contar con alguna instalación de *Git* en el sistema, para poderlo utilizar desde *RStudio*. Para utilizarlo, podemos auxiliarnos con la biblioteca `usethis`, que ofrece una amplia batería de comandos para interactuar con la plataforma de *Git* y con el servicio de Internet *Github*.

En *RStudio* hay ayuda para utilizar *Git* directamente. Para acceder a este apoyo deberás crear un proyecto que incluya el control de versiones como una característica activa. En la figura se ilustran las dos rutas para hacerlo en *RStudio*.

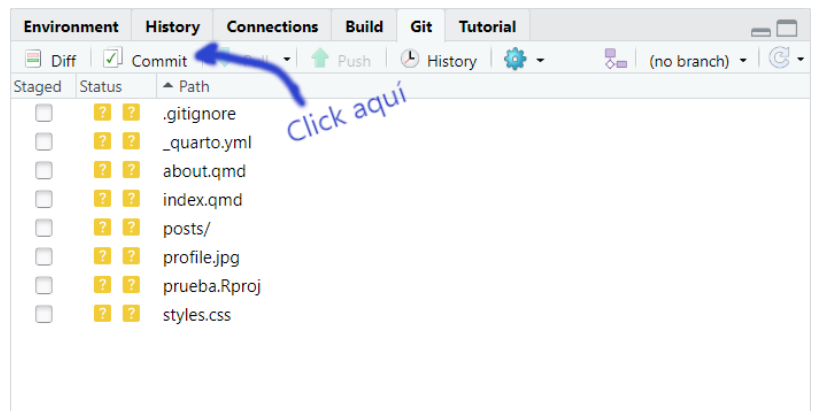


Evidentemente, si seguiste la ruta azul, tu repositorio ya existe en *Github*. Si optaste por la ruta verde, entonces deberás crear un repositorio allá. Parar hacerlo hay tres sencillos pasos adicionales que dar.

1. Definir la *rama* principal (se sugiere **main**, aunque también se usa **master**).



2. Haz un *commit* seleccionando todos los archivos que aparecen en la pestaña **Git**. Los puedes comentar anotado “inicial” o algo sugerente de ser el primer control registrado en el repositorio del proyecto.



3. Vincula tu repositorio local con un nuevo repositorio en *Github*. La forma más sencilla de hacer esto con *RStudio* es escribir, en la pestaña de **Consola** el comando de *R*:

```
usethis::use_github()
```

Una vez que todo está preparado la rutina de trabajo incluye hacer *commit* cada vez que convenga preservar alguna versión. De vez en cuando sincronizar con la copia maestra del repositorio en *Github*, primero obteniendo el estado actual con el comando *pull* y luego enviando la versión de tu equipo local con un *push*.

Algunos problemas que se nos presentaron.

1. *Git* no instalado. Esperábamos que la instalación de *Github desktop* instalara *Git* en una sola operación. Por alguna razón esto no fue así. Recurrimos a *Git for Windows* y lo instalamos directamente.
2. Datos de usuario de *Git* no registrado. Lo resolvimos con `Git bash`, que es una consola de comandos para operar el equipo que está vinculado con la instalación de *Git*.

```
git config --global user.name "Mona Lisa"
git config --global user.email "HER_EMAIL"
```

Verificar que tus datos quedaron registrados:

```
git config --global user.name
git config --global user.email
```

Sin embargo, también podríamos haber usado el siguiente comando de `usethis`

```
usethis::use_git_config(user.name = "Mona Lisa", user.email = "HER_EMAIL")
```

3. Acceso a *Github* pide registrar un **token** en el equipo. Lo resolvimos con ayuda de la biblioteca `usethis` Los comandos que ejecutamos en la pestaña de *consola* fueron:

```
usethis::create_github_token()
```

Esto lleva a la página de Github en la que hay que generar el **token**. Hay que responder las preguntas que te haga la página y apretar el botón respectivo. Aparecerá una nueva pantalla con el **token** que habrá que copiar al *portapapeles* (ctr-c en windows). En seguida hay que ejecutar este otro comando en la consola de *RStudio*

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

4. La operación de *Netlify* parecía no funcionar correctamente. Encontramos que el proyecto en *RStudio* no estaba completamente *renderizado*. Lo resolvimos activando **Render website** en la pestaña **Build** de *RStudio*.
5. Una manera de averiguar muchos detalles esenciales de tu proyecto es usar el siguiente comando. Producirá un reporte un poco largo. De particular interés es la sección *Active usethis project*. También será oportunidad de verificar que estés vacunado (evita en lo posible distribuir información confidencial, entre otras cosas).

```
usethis::git_sitrep()
```

Si no estas “vacunado”, conviene que lo hagas. Puedes encontrar más información al respecto con

```
? git_vaccinate
```

o yendo a la pestaña de ayuda y buscando este mismo comando.

```
usethis::git_vaccinate()
```

6. Encontré este [repo de AaronGullickson](#) que quiso compartir lo que considera comandos útiles de `usethis` para usuarios de *RStudio*.