Git: Configuración y primeros comandos

Emmanuel Arias

1. Configuración

Lo primero que necesitamos realizar una vez descargado e instalado Git es su configuración.

Podemos configurar variables en tres ambientes, sistema, global y local. Las variables se guardan en archivos.

- Cuando configuramos una variable en el ambiente sistema, esta queda disponible para todo el sistema. En sistemas Linux se guardan en `/etc/gitconfig` en sistemas Windows se guarda en `C:\ProgramData\Git\config`.
- Cuando se configura en el ambiente global, esto significa que está disponible para un usuario en específico. Es decir que esa configuración va a estar disponible para todos los proyectos del usuario. Estos datos se guardan en el archivo `~/.gitconfig` o `C:\Users\\.gitignore`
- Cuando configuramos utilizando el ambiente local, estas variables solo se encuentran disponible para un proyecto dónde fue configurado.

Para configurar utilizamos el comando "config", seguido por el argumento que indique en qué ambiente queremos guardar esa información.

Normalmente, configuramos Git utilizando el ambiente global, ya que de esta manera nos permitirá tener siempre disponible esta información para todos los proyectos que manejemos, en casos muy puntuales necesitaremos cambiar nuestra información en algún proyecto, por lo que usaremos en ese caso la configuración "local.

Primero, el comando tiene la siguiente forma:

```
git config --system nombre.variable "valor"
git config --global nombre.variable "valor"
git config --local nombre.variable "valor"
```

Como primer paso configuremos nuestro git con la información mínima que necesitamos para darle autoría a nuestros trabajos.

```
git config --global user.name "Emmanuel Arias"
git config --global user.email "eamanu@yaerobi.com"
```

Luego deberemos configurar un editor de texto que se abra cada vez que necesitemos hacer un "commit". Para ello, necesitamos configurar la variable core.editor seguido por el nombre del editor (esto funciona en Linux), o el path al ejecutable en Windows. Por ejemplo, yo configuro para que mi editor por defecto sea vim.

```
git config --global core.editor "vim"
```

1.1. Configuración ssh

Cuando usamos Github cada vez que sincronizamos con nuestro repositorio local, esto es, cada vez que envíamos cambios a Github o queremos traer información de Github (por ejemplo, clonar un repositorio, hacer un pull, que ya veremos que es eso) Necesitamos autenticarnos. Para autenticarnos, la forma más sencilla es escribiendo nuestro usuario y contraseña y eso es todo, pero para ellos siempre

tendremos que trabajar con el protocolo *HTTP*. La otra forma es utiliza *ssh* y de esa manera nos evitamos escribir a cada rato usario y contraseña, y bueno obviamente es más seguro.

Para ello, simplemente seguimos la documentación de Github y listo :)

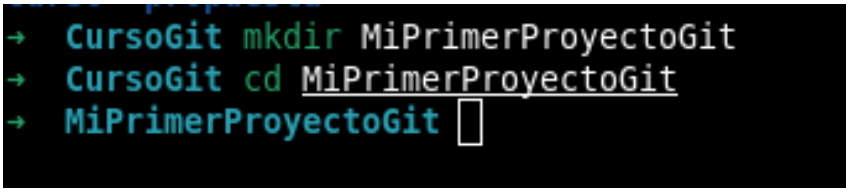
En este link podrá configurar Github con ssh en Windows, MacOS o Linux

2. Nuestro primer proyecto

Para crear nuestro primer proyecto vamos a utilizar el comando "init". Ya depende de cada uno y su forma de trabajar, pero yo recomiendo ser organizado y tener un directorio para guardar nuestros proyectos.

Entonces, vamos a crear nuestro primer proyecto. Primero debemos una carpeta con el nombre del proyecto, y vamos a acceder a él. Lo pueden hacer desde la GUI, o por comandos, yo lo haré por comandos pero es lo mismo.

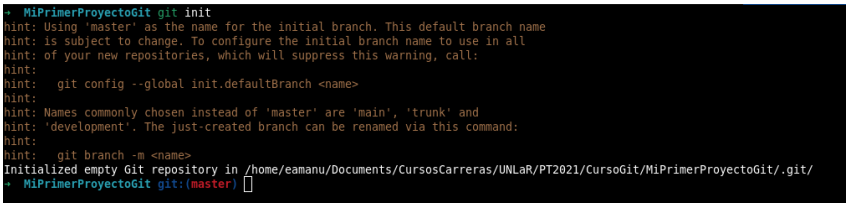
```
mkdir MiPrimerProyectoGit
cd MiPrimerProyectoGit
```



(Eso equivale a botón derecho, Nueva Carpeta. Luego F2 y escribir el nombre de la carpeta y por último presionar dos veces en la carpeta.)

Luego abriremos nuestra línea de comandos o Git Bash si estamos en Windows, y escribiremos

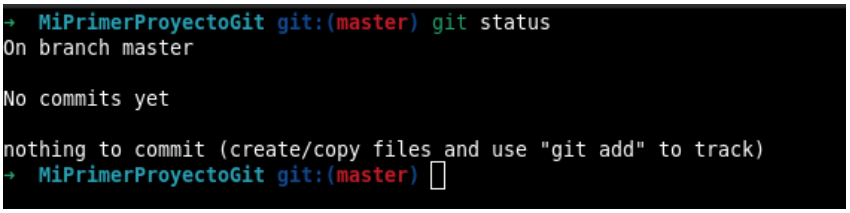
```
git init
```



Y listo! con esto ya tendremos nuestro proyecto creado. Ahora ya podemos empezar a trabajar.

Podremos verificar que está creado si usamos el comando

```
git status
```

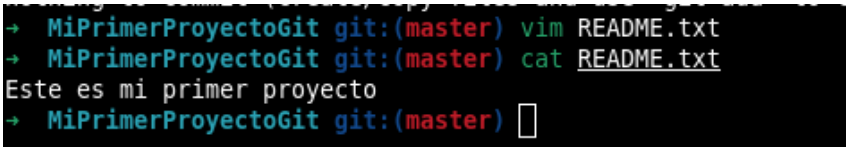


2.1. Agregando archivos al proyecto

Por lo pronto vamos a hacer algo muy simple, vamos a crear un archivo y vamos a escribir lo que sea, solo es un primer proyecto, ya tendremos tiempo para cosas más avanzadas.

Creamos un archivo como lo harían normalmente, ojo, mejor si es texto plano, para evitar algunas cuestiones que no vienen al caso por ahora.

En mi caso, solo pondré en ese archivo "Este es mi primer proyecto", y el nombre? cualquiera, puede ser... README.txt.



Ahora hacemos un `git status` para ver cómo avanzamos:

Bueno aparece en rojo nuestro archivo "README.txt". Eso es bueno

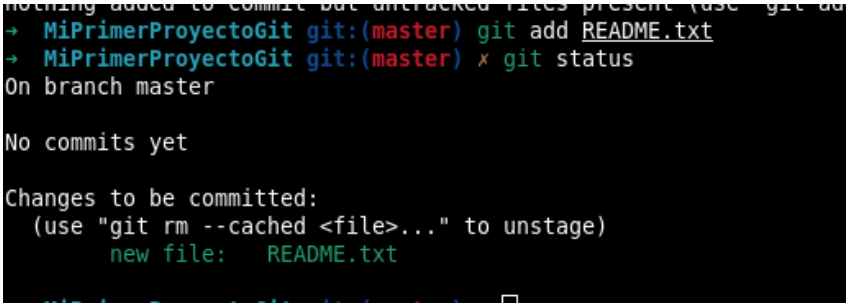
Puntos a tener en cuenta hasta ahora:

- Todos los comandos que estamos viendo, init, status, etc y los que veremos en breve, ya tendremos más tiempo de estudiarlos en otras unidades, primero hagamos funcionar Git, y luego veamos los detalles.
- Cuando hicieron el último `git status` vieron más archivo o carpetas que README.txt: *Houston, we've a problem.* Acá hay algo mal, y deberán volver al inicio.

Se acuerdan de la clase de las etapas o "stages"de Git? Bueno, ahora README.txt está en Untracked, no está en Git todavía. Vamos a agregarlo, para ello ejecutamos

```
git add README.txt
```

Ahora si hacemos git status aparecerá en otro color:

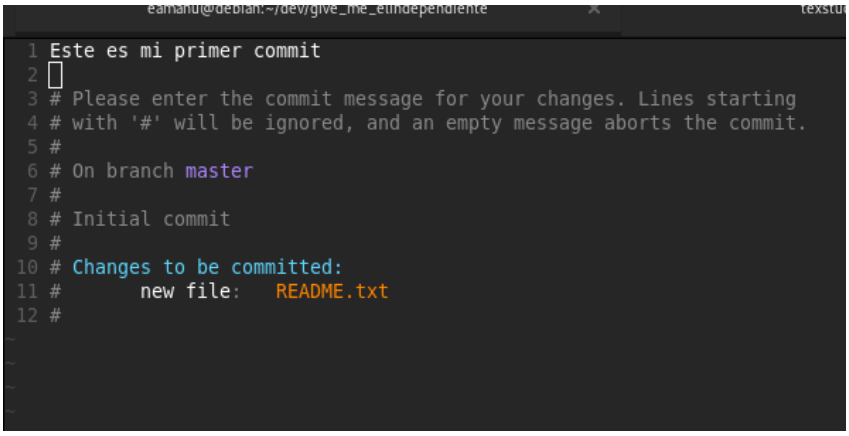


Ahora está en Stagging. Vamos a agregarlo en el repositorio para que ya quede guardada en la historia de Git.

Para ello hacemos

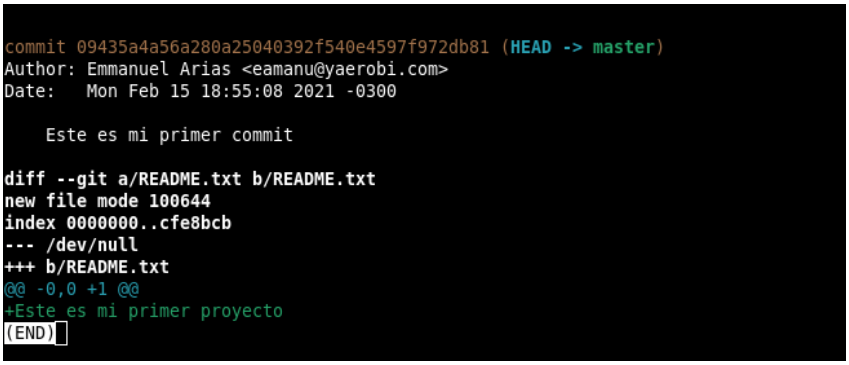
```
git commit
```

Esto nos abrirá el editor que habíamos configurado en 1 y escribimos un mensaje. Una vez escrito el mensaje, lo guardan y salen del editor. En mi caso lo hago en vim (si por esas casualidades, utilizaron vim y no saben salir del editor, aprieten el botón ESC y luego escriban `:wq` y presionen Enter, y listo.), y se ve así:



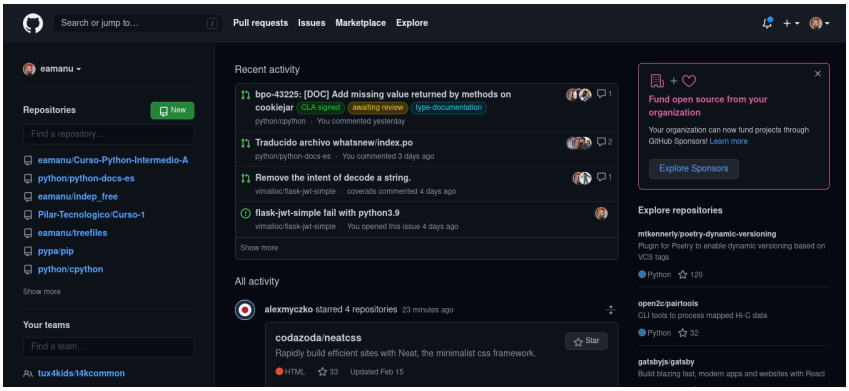
Si ejecutan status verán que ya no hay nada nuevo, así que *voilà*, tenemos nuestro primer commit realizado!!!.

¿Cómo lo vemos? Pueden ejecutar `git show` o `git log`.

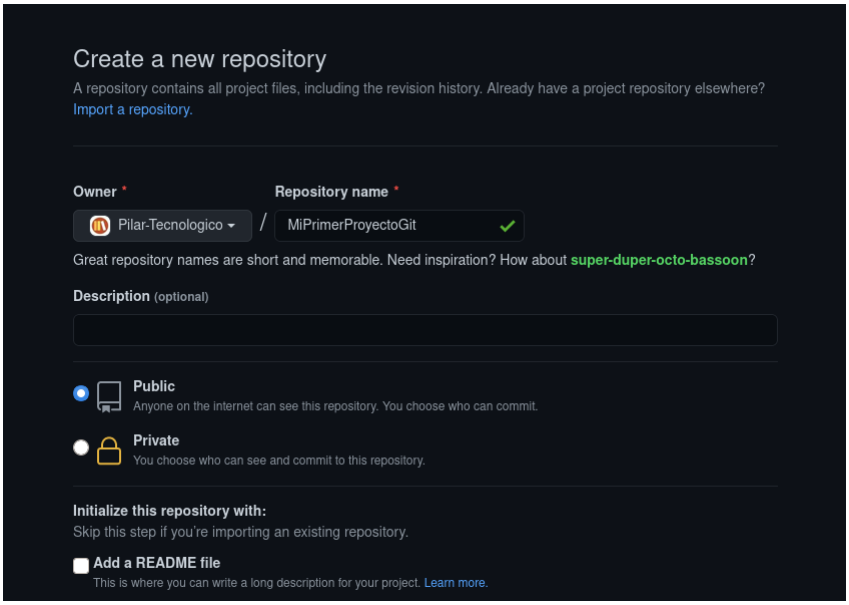


3. Vamos a Github

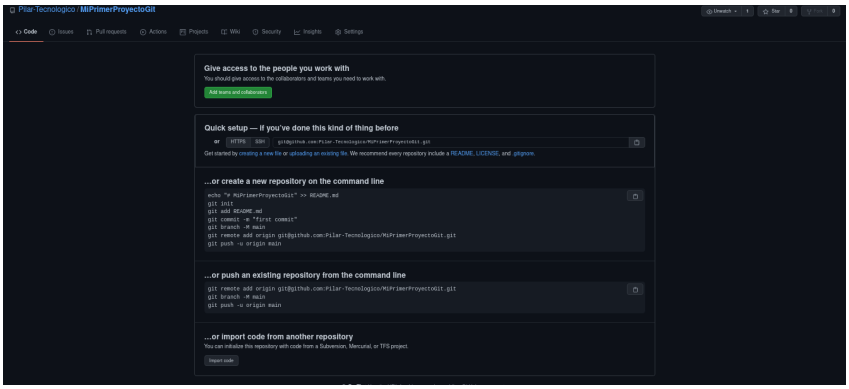
Vamos a ingresar a Github, y creamos un nuevo repositorio, haciendo clic en el botón que aparece a la izquierda que dice “Nuevo”:



Luego le ponemos un nombre al proyecto, mantengamos el mismo. Hacemos clic en "Public". Y en la última sección que nos pregunta si queremos "inicializar el proyecto con"**NO seleccionamos ninguno**.



Apretamos en el botón crear repositorio. Vamos a tener algo similar a esto:



Veremos que ya nos da mucha información de lo que debemos hacer a continuación lo cual es bueno, nosotros ya creamos un proyecto, así que tenemos que hacer lo que nos dice la sección "push un proyecto existente desde la línea de comandos".

Antes que nada, si ya hemos configurado nuestro Github para que funcione con ssh (1.1) elegimos la opción ssh que aparece arriba en la página, si no lo hicimos, tendremos que seleccionar la opción http.

Simplemente copiamos línea a línea y eso es todo.

```
→  MiPrimerProyectoGit git:(master) git remote add origin git@github.com:Pilar-Tecnologico/MiPrimerProyectoGit.gi
→  MiPrimerProyectoGit git:(master) git branch -M main
→  MiPrimerProyectoGit git:(main) git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 905 bytes | 905.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Pilar-Tecnologico/MiPrimerProyectoGit.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
→  MiPrimerProyectoGit git:(main)
```

Ahora si refrescamos la página, veremos que nuestro código ya está en Github.