Taller Git Hub – Data Science

En este taller se utilizará la herramienta de escritorio GitHub Desktop para aprender las funcionalidades básicas de Git:

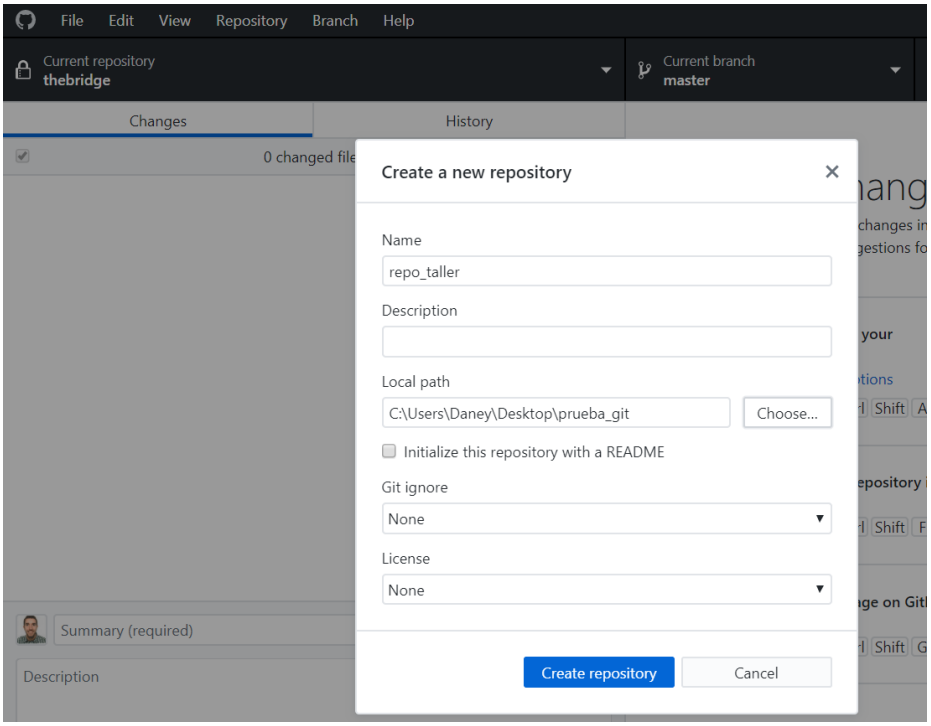
1. **git init** - Crear un repositorio
2. **git commit -** Guardar diferentes versiones de tu repositorio
3. Publicar tu repositorio en GitHub
4. **git push** – Versiona de nuevo tu programa y sube los cambios
5. Eliminar commits
6. **git clone** - Clonar un repositorio
7. **git pull** - Descargar los cambios del repositorio remoto

Necesitas una cuenta de GitHub. Es gratuita

1. Crear un repositorio

Elige una ruta de tu ordenador donde vayas a crear tu primer repositorio. En mi caso: "C:\Users\Daney\Desktop\prueba_git". Carpeta vacía.

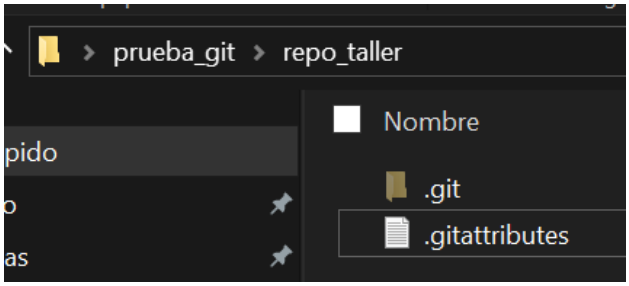
Desde GitHub Desktop vamos a file -> New Repository. Le ponemos un nombre al repositorio. Y en el *Local path*, la ruta que hemos elegido. El resto lo podemos dejar como está.



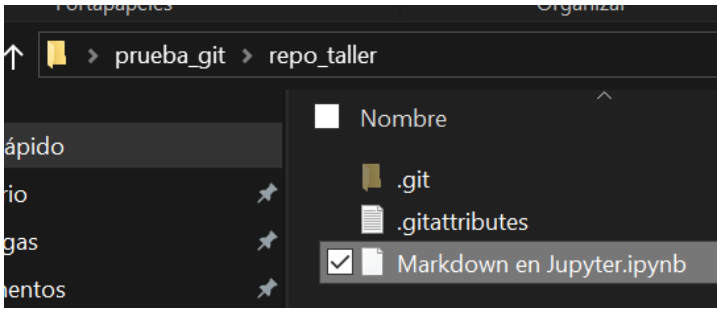
Ya tienes el repositorio creado en tu ordenador, en local. GitHub Desktop ha creado una carpeta con el nombre que hayas puesto, dentro de la ruta que indicaste. Si quieres crear un repositorio a partir de una ruta donde ya tienes archivos, en vez de File -> New Repository, tendrías que ir a File -> Add local repository…

2. Guardar diferentes versiones en tu repositorio

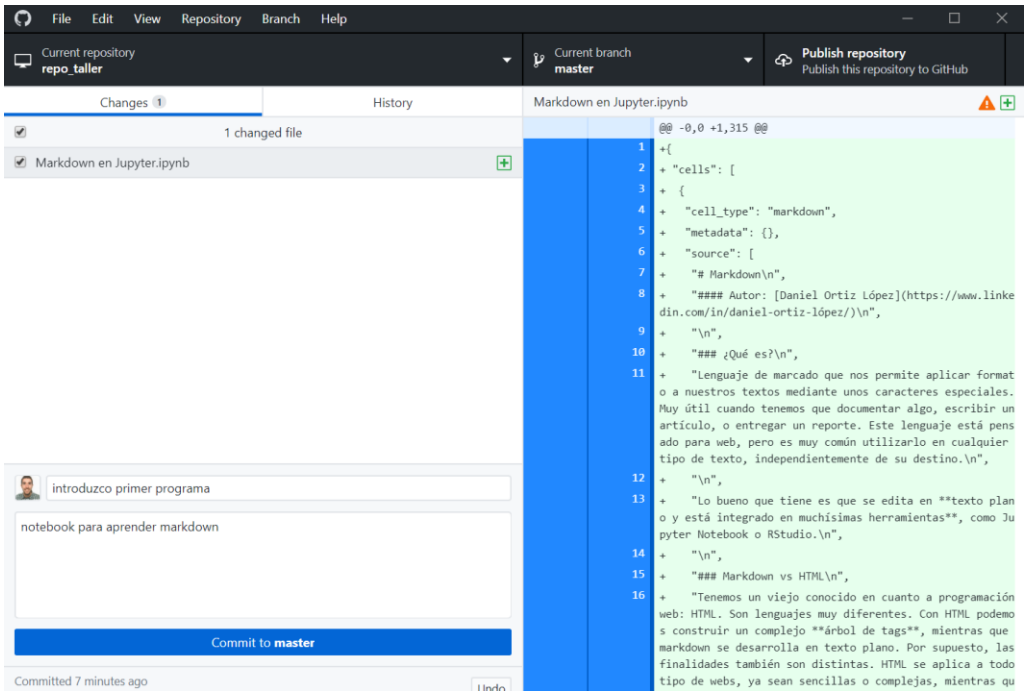
Deberías tener una carpeta con los archivos de la siguiente imagen. Son archivos de configuración de Git.



Crea tu primer programa dentro del repositorio. Por comodidad, copio y pego uno ya creado.



Ahora ve a Git y verás que ha cambiado. Te dice que hay cambios en un archivo. Bien, toca "sacarle una foto" a nuestro repositorio, es decir, realizar el primer *commit*. Introduce un título y una descripción al *commit*. Esto es muy importante porque si en un futuro queremos volver a versiones anteriores de nuestro software, tendremos que saber qué hicimos en cada *commit*. Haz click en *commit to master*.

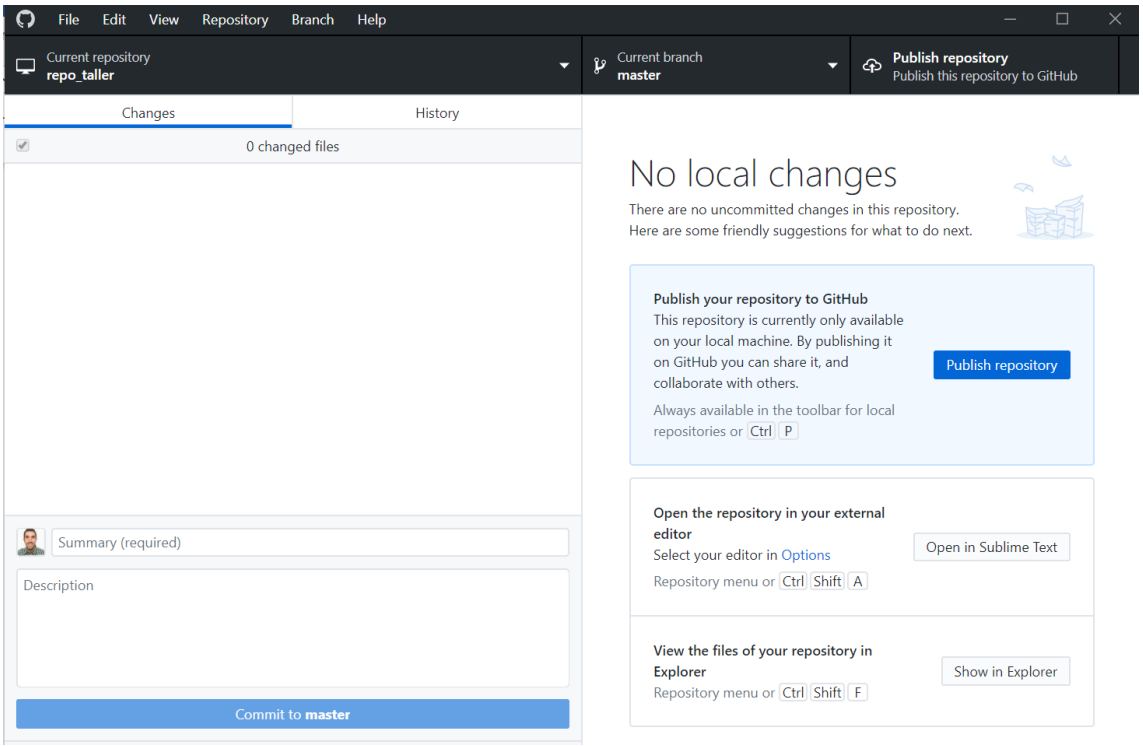


Git tiene la opción de dividir el desarrollo en varias ramas (o *branches*). La principal se llama master, y durante este taller únicamente trabajaremos con la rama principal.

3. Publicar tu repositorio en GitHub

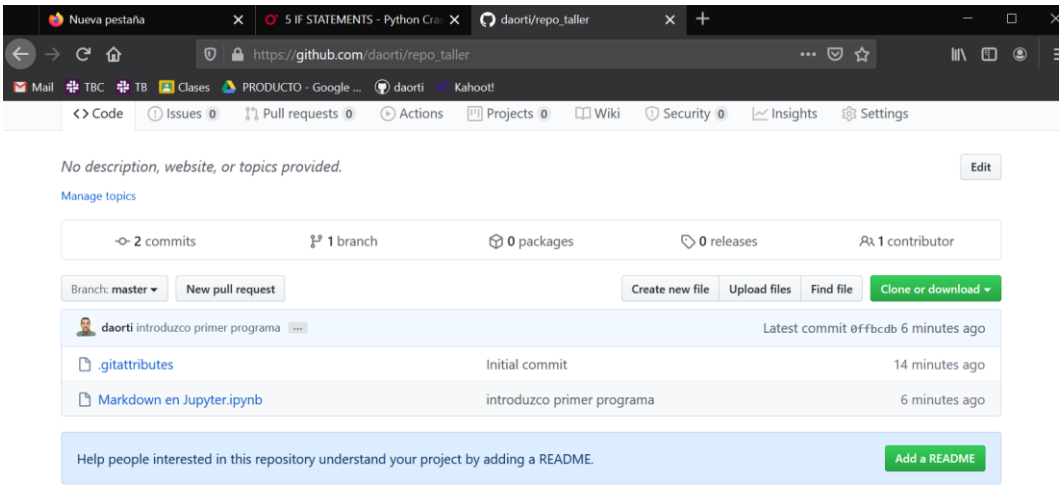
Ya tenemos un repositorio creado en local. Y tenemos una primera versión. Ahora cada vez que hagas cambios en el repositorio, GitHub Desktop te sugerirá hacer un commit nuevo, para guardar esos cambios.

Ahora vamos a trasladar toda esta lógica de versiones, a la nube, a GitHub. Tras realizar el primer commit, debería aparecer un cartel de que ya no hay cambios, sugiriendo que lo publiquemos en GitHub. Le decimos que si.



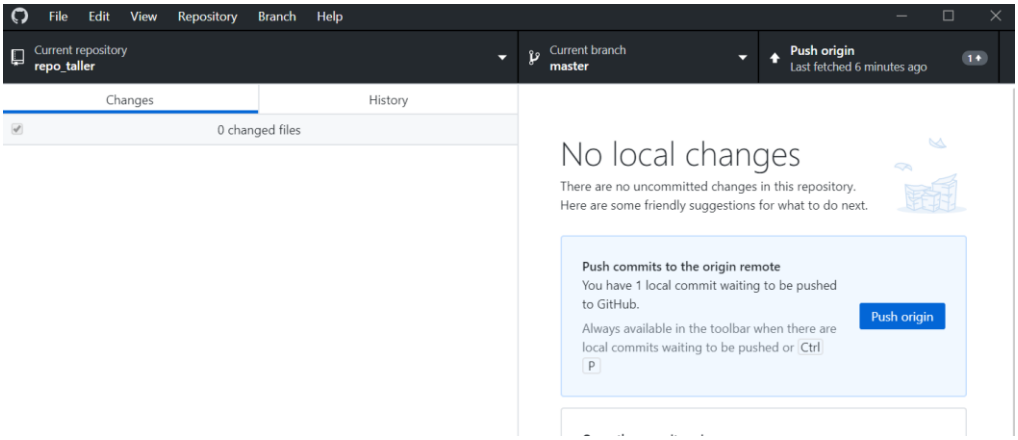
Ponemos un nombre al repositorio (deja el que esté), descripción y quitamos lo de que sea privado.

¡Accede a GitHub y comprueba que ya tienes tu primer repositorio, versionado y subido en la nube! Desde GitHub Desktop puedes abrir GitHub simplemente con (ctrl + shift) + G.

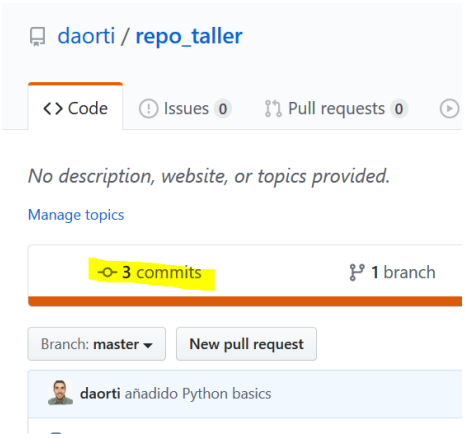


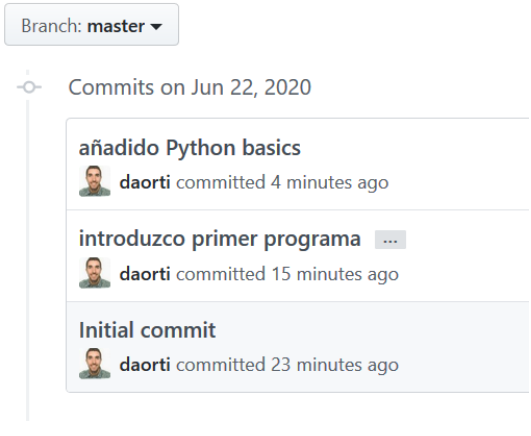
4. Versiona de nuevo tu programa y sube los cambios

Añade otro archivo a tu repositorio y haz otro *commit*. Ahora ya no te sugerirá publicar el repositorio, sino que te pide subir los cambios, un *push*. Por supuesto, puedes hacer varios commits y subir después los cambios.



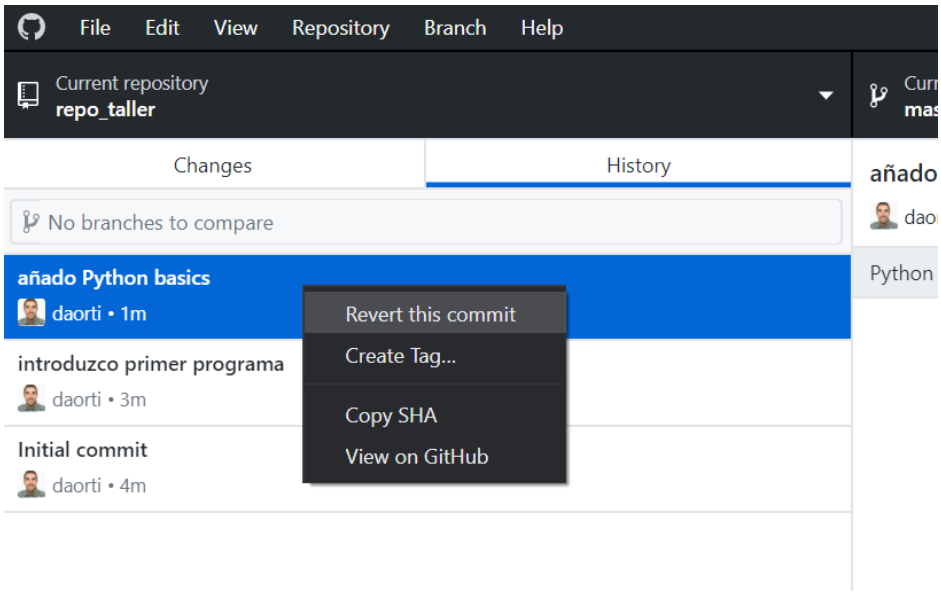
Vete ahora al repositorio de GitHub y selecciona la pestaña de commits. Verás el histórico de versiones del programa. Y no solo eso, sino que tendrás al usuario que hizo cada versión, cuándo la hizo y una descripción de por qué se hizo.





5. Eliminar commits

Somos muy ordenados si sacamos versiones de nuestro software, pero no sirve de nada si no aprovechamos esta funcionalidad. Vamos a suponer que el ultimo cambio que hicimos no fue bueno, y queremos volver al commit anterior. Desde GitHub Desktop -> History -> Selecciono el *commit* a borrar (en este caso "añado Python basics") -> botón derecho -> *Revert this commit*



De esta forma eliminamos el ultimo commit, y nos quedamos con la version inmediatamente anterior. Si queremos hacer cosas más específicas como ir a un commit en concreto, tendremos que hacerlo desde git.

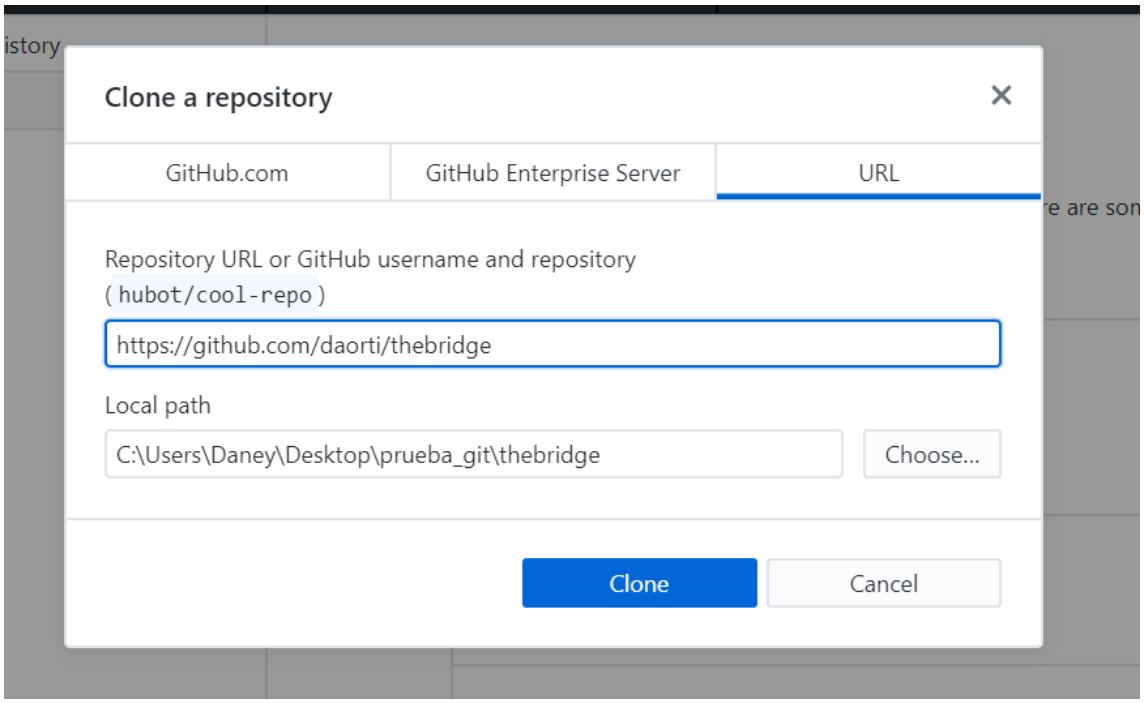
6. Clonar un repositorio

Necesitamos tres cosas:

- Ruta del ordenador donde queramos clonar el repo
- Tener acceso al repo
- Link de GitHub del repo

Lo primero, seleccionamos ruta en el ordenador (C:\Users\Daney\Desktop\prueba_git). Lo segundo, pedir acceso al repo en el caso de que sea privado, y por último, conseguir la ruta del repositorio (preguntar a los profesores).

Ya con todo esto, podemos ir a GitHub Desktop, File -> Clone Repository -> URL. Pegamos la URL del repositorio y seleccionamos en local path, la rutal del ordenador donde queremos dejar el repositorio. Seguidamente, hacemos clik en Clone.



Ahora en la carpeta deberíamos tener un nuevo repositorio con todos los archivos del curso.

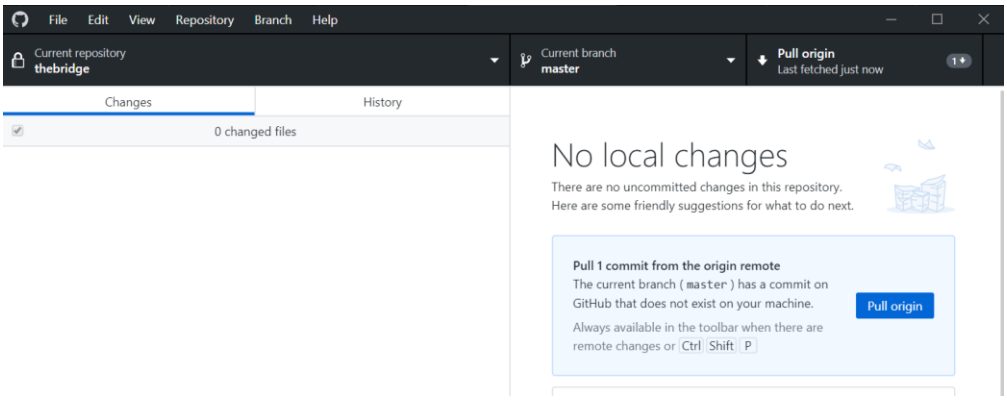
IMPORTANTE. Se recomienda ir actualizado esta carpeta para bajarse todos los materiales del curso, pero NO usarla para guardar tu propio material, ya que vas a tener conflictos en las versiones de los archivos. Lo recomendable es que desarrolles tus propios notebooks en otra ruta y que dejes esta únicamente para que se actualice con los últimos cambios que haya en el repositorio de los profesores. Si hemos hecho algo mal, siempre tenemos la opción de clonar de nuevo el repo 😊.

7. Descargar los cambios del repositorio remoto

Solo nos queda por ver cómo se actualiza el repositorio en nuestro ordenador, con los últimos cambios.

Lo primero, ¿cómo se que hay cambios? Hay que darle a refrescar en *Fetch origin* para comprobar si hay una nueva versión.

En caso de que la haya, nos sugerirá que hagamos un **pull**, para descargárnosla. Clickamos en pull y listo. Repo actualizado.


File
Edit
View
Repository
Branch
Help
Current repository
thebridge
Current branch
master
Pull origin
Last fetched just now
1
Changes
History
0 changed files
No local changes
There are no uncommitted changes in this repository.
Here are some friendly suggestions for what to do next.
Pull 1 commit from the origin remote
The current branch (master) has a commit on GitHub that does not exist on your machine.
Pull origin
Always available in the toolbar when there are remote changes or Ctrl Shift P