

# Ciencia de Datos

- Módulo 2

Git



## Control de versiones

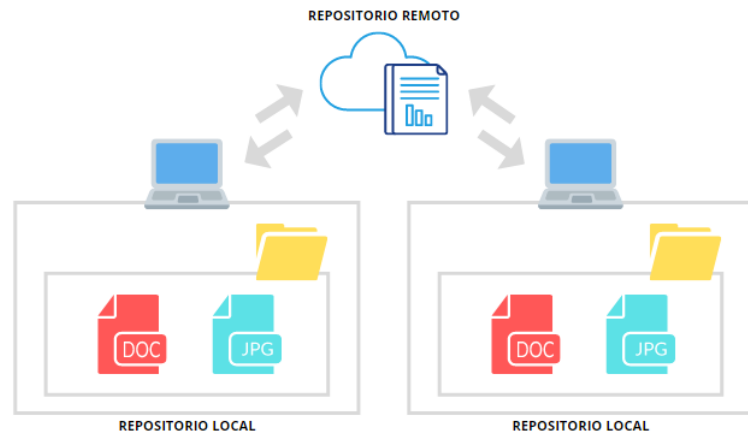
Un sistema de control de versiones sirve para rastrear cambios en archivos, es decir, cada vez que se modifica un archivo, el sistema de control de versiones registra y guarda cada cambio y quien lo hizo. Esto le permite al equipo seguir el estado y evolución de los archivos, y, en caso de necesitarlo restaurar a una versión anterior de algún archivo en cualquier momento. El sistema más usado es Git.



# Git

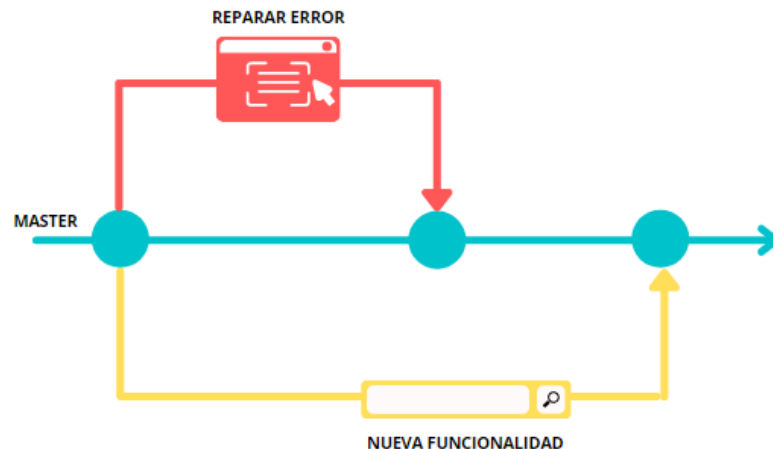
Git se basa en usar un repositorio remoto, generalmente en la nube, y copias del mismo en cada una de las computadoras de los miembros del equipo, estos los llamamos repositorios locales.

Esto permite a los desarrolladores trabajar en el mismo archivo de forma simultánea, sin el peligro de sobrescribir el trabajo de otro.



## Git - ramas

Otra funcionalidad de Git es el manejo de ramas. Estas son “copias” del repositorio, sirven para hacer cambios o experimentar en el proyecto sin miedo a dañar el código fuente. Una vez que terminemos de desarrollar lo que queríamos hacer en la rama, podemos fusionarla (merge) con la rama principal, o simplemente descartar los cambios.



## Control de versiones - GitHub

Una de las implementaciones de Git más populares, sobre todo para proyectos de código abierto, implementaciones de bibliotecas, etc. es GitHub.

GitHub ofrece la posibilidad de hacer repositorios en la nube, configuración de usuarios, control de versiones y manejo de ramas entre otras cosas sin costo, para usarlo solo debemos registrarnos de forma gratuita.

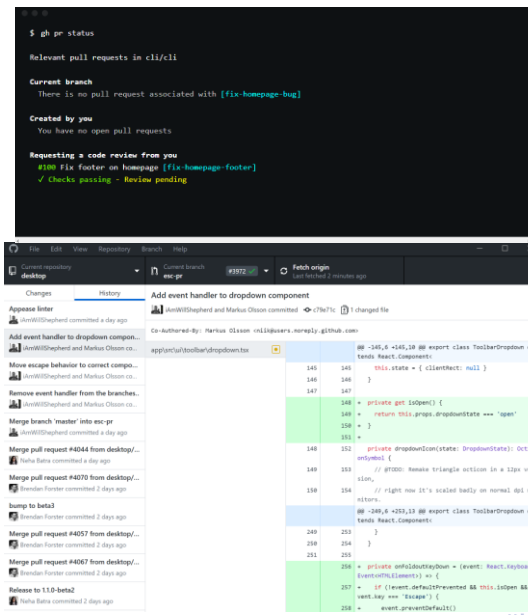
GitHub no se limita solo a Python puedes almacenar código en cualquier lenguaje, y aún más, también se pueden almacenar archivos en cualquier formato, pero se recomienda solo almacenar archivos de código.



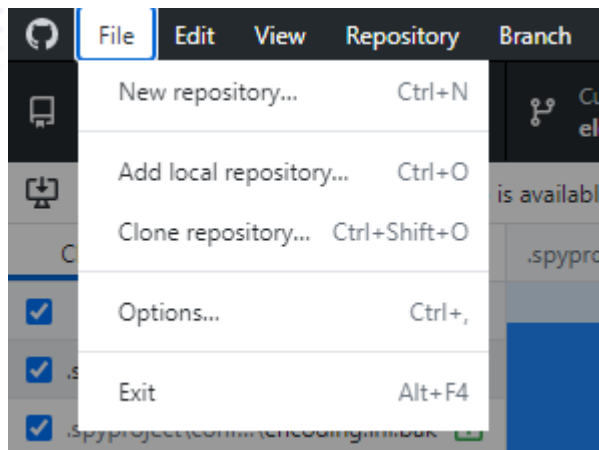
## Control de versiones - GitHub

GitHub, así como otras implementaciones de Git, se pueden manejar de dos formas distintas.

- **CLI:** Es la forma más básica y con mayor control, pero puede resultar un poco más compleja. CLI se refiere de Command Line Interface, que quiere decir interfaz por línea de comando.
- **GUI:** Viene de Graphic User Interface, o interfaz gráfica de usuario, en el caso de GitHub se llama GitHub Desktop, en general tienen todas las funcionalidades básicas de Git, muy recomendable si no tienen experiencia en el uso de la terminal.



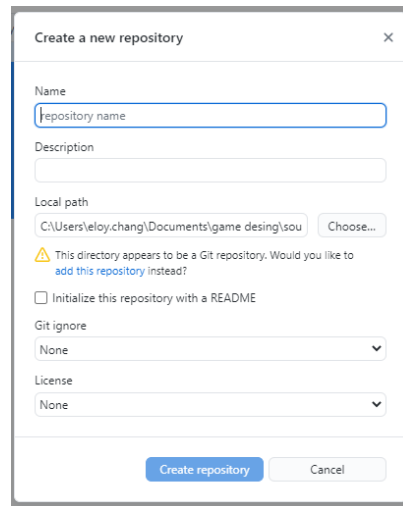
## Control de versiones - GitHub



Una vez que tenemos GitHub desktop instalado e iniciado sesión, podemos crear un nuevo repositorio en el menú File, opción New repository.

Luego nos aparece una ventana en la que debemos colocar el nombre del repositorio (asociado al tema a

tratar en el proyecto), una breve descripción opcional y la dirección de la carpeta local en que se guardaran los archivos.



## Control de versiones - GitHub

¡Listo! Ya creamos nuestro primer repositorio, sin embargo solo lo tenemos localmente, si queremos publicarlo en la nube de GitHub debemos darle a Publish repository, donde nos aparece una ventana para seleccionar de nuevo el nombre del repo (en la nube), la descripción y si queremos que sea privado o público, presionamos Publish repository y listo.

