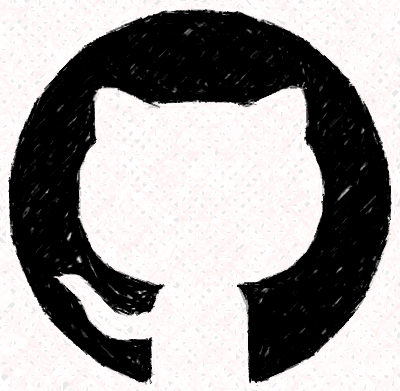
Tipos de comandos para usar en

GitHub



Los comandos de Git son herramientas fundamentales para interactuar con GitHub, una plataforma de alojamiento de repositorios de código. Aquí tienes una breve introducción a algunos comandos comunes:

1. **git init**: Inicializa un repositorio Git en un directorio local.
2. **git clone**: Descarga un repositorio de GitHub a tu computadora.
3. **git add**: Añade cambios al área de preparación (staging area) para ser confirmados.
4. **git commit**: Guarda los cambios en el repositorio local junto con un mensaje que describe las modificaciones.
5. **git push**: Sube los cambios confirmados a un repositorio remoto en GitHub.
6. **git pull**: Obtiene y fusiona cambios desde un repositorio remoto a tu repositorio local.
7. **git status**: Muestra el estado actual de los archivos en el repositorio (archivos modificados, añadidos, o sin seguimiento).
8. **git branch**: Lista, crea o elimina ramas (branches).
9. **git merge**: Fusiona cambios de una rama a otra.
10. **git checkout**: Cambia entre ramas o restaura archivos a una versión anterior.

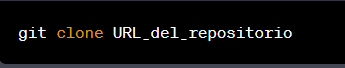
Estos son los comandos importantes los más básicos los cuales permiten al usuario gestionar versiones de cada ordenador y acoplara con una versión adaptable, permitir colaborar con otros desarrolladores, y controlar el flujo de trabajo en un repositorio Git alojado en GitHub.

A continuación, mostrare su sintaxis preferiblemente en lenguaje coloquial con información detallada de cada comando junto con su sintaxis , importante para que se estime pueda alcanzar a facilitar la comprensión al lector u usuario que posea este documento:

1. **git init**: Imagina que estás configurando un radar para seguir todos los cambios en una carpeta. Con esto, le dices a tu computadora que comience a mantener un registro de todo lo que pase en esa carpeta. Es como decirle a tu máquina: "¡Hey, a partir de ahora, estaré vigilando todos los cambios aquí!"



1. **git clone**: Este comando es como hacer una fotocopia de un proyecto que está en línea. Te permite descargar una copia exacta del proyecto de GitHub a tu computadora para que puedas trabajar en él como quieras, ¡sin preocuparte por dañar la versión original en línea!



1. **git add**: Es como elegir qué cosas quieres guardar de todas las modificaciones que has hecho en tu carpeta. Digamos que estás ordenando tu habitación y decides qué cosas guardar en tu armario.



1. **git commit**: Una vez que has decidido qué guardar, este comando es como tomar una foto de tus cosas ordenadas y escribir una nota sobre lo que has cambiado. Es como decirle a tu computadora: "¡He hecho estos cambios y aquí está la explicación de lo que hice!"



1. **git push**: Después de tomar esa foto, ahora quieres mostrarla a tus amigos. Con este comando, subes esa foto a tu cuenta de GitHub para que otros puedan ver tus cambios.



1. **git pull**: Si tus amigos han hecho cambios en la foto que todos comparten, con este comando puedes actualizar tu versión para ver esas modificaciones. Es como obtener la última versión de la foto grupal.



1. **git status**: ¿Te has preguntado qué cosas has ordenado y qué cosas faltan por guardar? Este comando te dice exactamente eso. Muestra una lista de todas las cosas que has cambiado y te dice si están listas para ser guardadas o no.



1. **git branch**: Imagina que estás trabajando en un proyecto y quieres probar algo nuevo sin estropear lo que ya tienes. Con este comando, creas caminos separados donde puedes experimentar sin afectar la versión principal de tu proyecto.



1. **git merge**: Una vez que has experimentado lo que querías en esos caminos separados y estás feliz con los cambios, este comando es como unir esos caminos al proyecto principal. Juntas todas esas pruebas en una sola versión.



1. **git checkout**: Imagina que te has probado un sombrero nuevo pero decides que el sombrero antiguo era mejor. Con este comando, puedes cambiar entre diferentes versiones del proyecto o volver a una versión anterior como si cambiaras de sombrero.

