



Comandos Básicos de Git

Git es un sistema de control de versiones de código abierto. Se utiliza para llevar un registro de los cambios realizados en un proyecto y permitir a los desarrolladores colaborar y trabajar en el mismo código de manera eficiente.

Git permite a los desarrolladores guardar diferentes versiones del código en un repositorio local y luego enviarlas al repositorio remoto para compartirlas con otros miembros del equipo. Los cambios pueden ser revisados y fusionados de manera controlada, lo que ayuda a prevenir conflictos y errores.

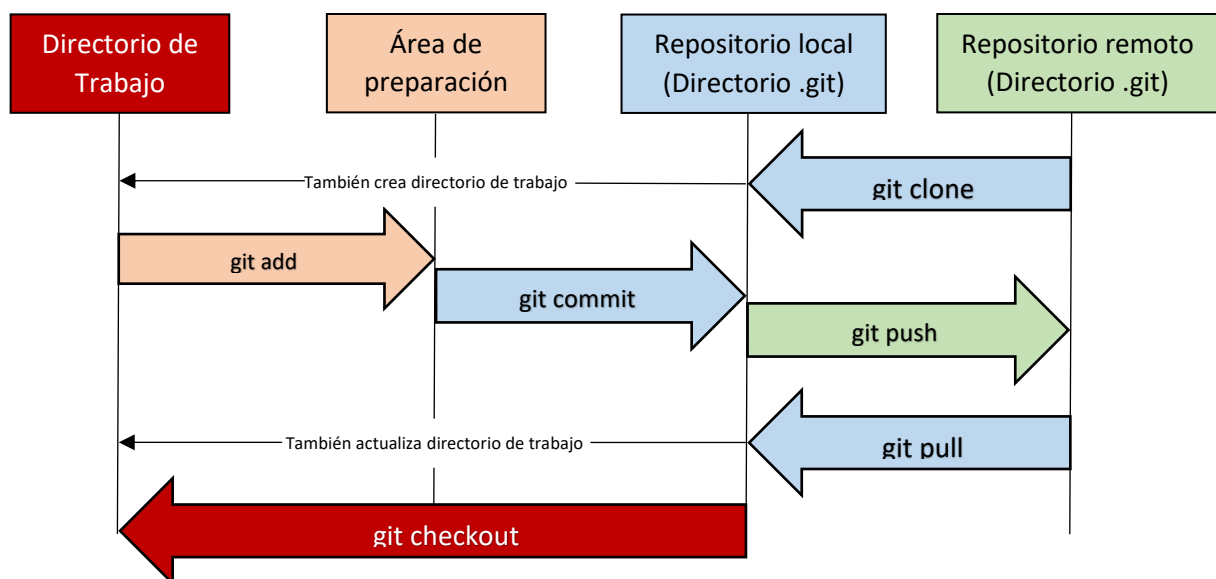
Zonas

Dentro de Git existe lo que se conoce como *las tres zonas*, estas zonas son la estructura base de Git y es esta forma en la que se divide este sistema de control de versiones lo que lo hace diferente de los demás.

El directorio de Git es donde se almacenan los metadatos y la base de datos de objetos para tu proyecto. Es la parte más importante de Git, y es lo que se copia cuando clonas un repositorio desde otra computadora.

El directorio de trabajo es una copia de una versión del proyecto. Estos archivos se sacan de la base de datos comprimida en el directorio de Git, y se colocan en disco para que los puedas usar o modificar.

El área de preparación es un archivo, generalmente contenido en tu directorio de Git, que almacena información acerca de lo que va a ir en tu próxima confirmación. A veces se le denomina índice ("index"), pero se está convirtiendo en estándar el referirse a ella como el área de preparación.





1. `git init`: Este comando se utiliza para inicializar un nuevo repositorio Git en una carpeta específica.
2. `git status`: Este comando se utiliza para ver el estado actual del repositorio, incluyendo los archivos modificados y los que están en el área de preparación.
3. `git add`: Este comando se utiliza para agregar archivos al área de preparación (staging area) para su próximo commit.
4. `git commit`: Este comando se utiliza para guardar los cambios en el repositorio local.
5. `git push`: Este comando se utiliza para enviar los cambios del repositorio local al repositorio remoto.
6. `git clone`: Este comando se utiliza para clonar un repositorio existente en una carpeta local.
7. `git pull`: Este comando se utiliza para recuperar los cambios del repositorio remoto y fusionarlos con el repositorio local.