

Comandos de GIT más usados

Aquí hay algunos comandos básicos de GIT que debes conocer:

- **git init** creará un nuevo repositorio local GIT. El siguiente comando de Git creará un repositorio en el directorio actual:

```
git init
```

- Como alternativa, puedes crear un repositorio dentro de un nuevo directorio especificando el nombre del proyecto:

```
git init [nombre del proyecto]
```

- **git clone** se usa para copiar un repositorio. Si el repositorio está en un servidor remoto, usa:

```
git clone nombredeusuario@host:/path/to/repository
```

- A la inversa, ejecuta el siguiente comando básico para copiar un repositorio local:

```
git clone /path/to/repository
```

- **git add** se usa para agregar archivos al área de preparación. Por ejemplo, el siguiente comando de Git básico indexará el archivo temp.txt:

```
git add <temp.txt>
```

- **git commit** creará una instantánea de los cambios y la guardará en el directorio git.

```
git commit -m "El mensaje que acompaña al commit va aquí"
```

- Ten en cuenta que los cambios confirmados no llegarán al repositorio remoto.
- **git config** puede ser usado para establecer una configuración específica de usuario, como el email, nombre de usuario y tipo de formato, etc. Por ejemplo, el siguiente comando se usa para establecer un email:

```
git config --global user.email tuemail@ejemplo.com
```

- La opción -global le dice a GIT que vas a usar ese correo electrónico para todos los repositorios locales. Si quieres utilizar diferentes correos electrónicos para diferentes repositorios, usa el siguiente comando:

```
git config --local user.email tuemail@ejemplo.com
```

- **git status** muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser preparados o confirmados.

```
git status
```

- **git push** se usa para enviar confirmaciones locales a la rama maestra del repositorio remoto. Aquí está la estructura básica del código:

```
git push origin <master>
```

- Reemplaza <master> con la rama en la que quieres enviar los cambios cuando no quieras enviarlos a la rama maestra.
- **git checkout** crea ramas y te ayuda a navegar entre ellas. Por ejemplo, el siguiente comando crea una nueva y automáticamente se cambia a ella:

command `git checkout -b <branch-name>`

- Para cambiar de una rama a otra, sólo usa:

`git checkout <branch-name>`

- **git remote** te permite ver todos los repositorios remotos. El siguiente comando listará todas las conexiones junto con sus URLs:

`git remote -v`

- Para conectar el repositorio local a un servidor remoto, usa este comando:

`git remote add origin <host-or-remoteURL>`

- Por otro lado, el siguiente comando borrará una conexión a un repositorio remoto especificado:

`git remote <nombre-del-repositorio>`

- **git branch** se usa para listar, crear o borrar ramas. Por ejemplo, si quieres listar todas las ramas presentes en el repositorio, el comando debería verse así:

`git branch`

- Si quieres borrar una rama, usa:

`git branch -d <branch-name>`

- **git pull** fusiona todos los cambios que se han hecho en el repositorio remoto con el directorio de trabajo local.

`git pull`

- **git merge** se usa para fusionar una rama con otra rama activa:

```
git merge <branch-name>
```