

Configuración inicial

Comenzamos configurando la información del usuario

```
git config --global user.name "John Doe"

git config --global user.email johndoe@example.com
```

Crea un repositorio nuevo

Para crear un nuevo repositorio de git

```
git init
```

Hacer checkout a un repositorio

Crear una copia local del repositorio ejecutando

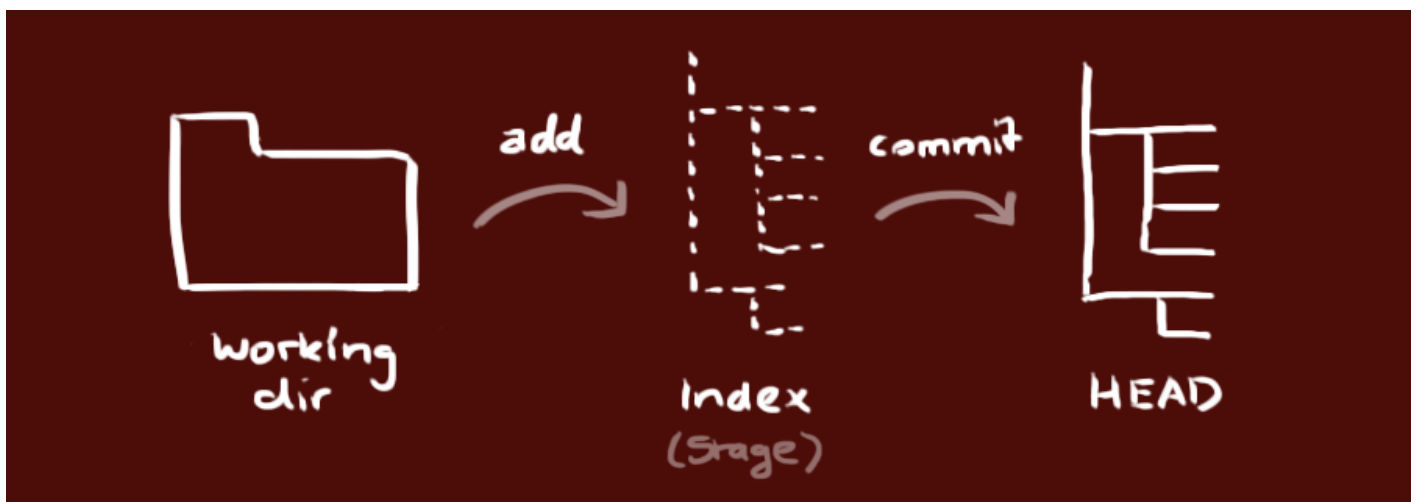
```
git clone /path/to/repository
```

Si utilizas un servidor remoto será algo como:

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

Flujo de trabajo

Tu repositorio local esta compuesto por tres "árboles" administrados por git. El primero es tu Directorio de trabajo que contiene los archivos, el segundo es el Index o área de preparación que actúa como una zona intermedia, y el último es el HEAD que apunta al último commit realizado.



Add & Commit

Puedes registrar cambios (añadirlos al área de preparación o Index) usando

```
git add <filename>  
git add .
```

Este es el primer paso en el flujo de trabajo básico. Para hacer commit a estos cambios usa

```
git commit -m "Commit message"
```

Ahora el archivo está incluido en el HEAD, pero aún no en tu repositorio remoto.

Configurar un remoto

Para configurar un directorio remoto donde conservar una copia

```
git remote add origin https://.... .git  
git branch -M main
```

Envío de cambios

Tus cambios están ahora en el HEAD de tu copia local. Para enviar estos cambios a tu repositorio remoto ejecuta

```
git push -u origin main
```

Reemplaza master por la rama a la que quieres enviar tus cambios.

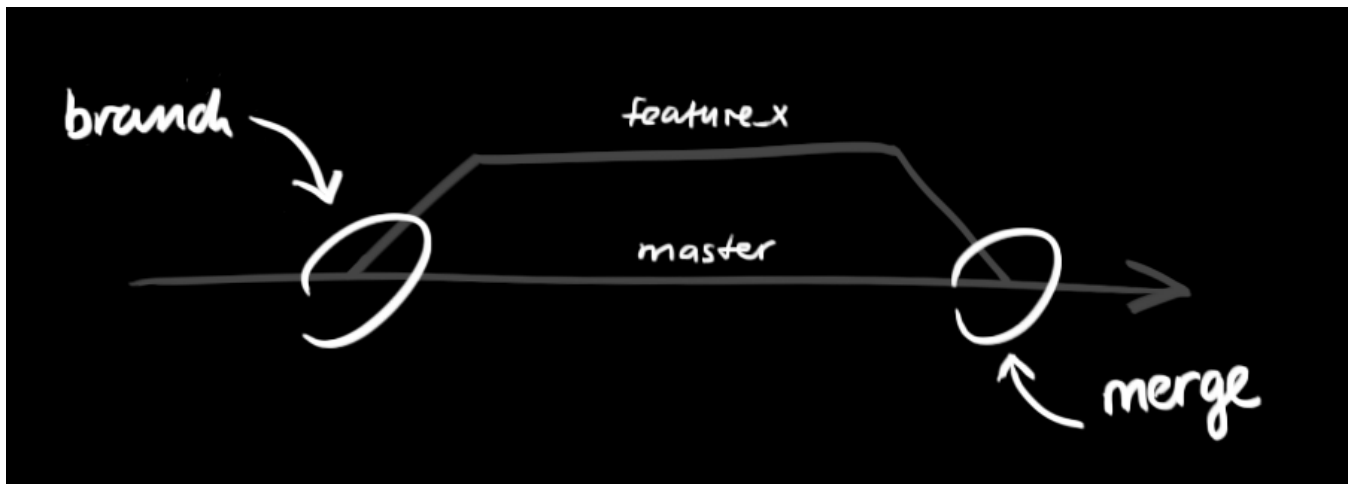
Si no has clonado un repositorio ya existente y quieres conectar tu repositorio local a un repositorio remoto, usa

```
git remote add origin <server>
```

Ahora podrás subir tus cambios al repositorio remoto seleccionado.

Ramas

Las ramas son utilizadas para desarrollar funcionalidades aisladas unas de otras. La rama master es la rama "por defecto" cuando creas un repositorio. Crea nuevas ramas durante el desarrollo y fusiónalas a la rama principal cuando termines.



Crea una nueva rama llamada "feature_x" y cámbiate a ella usando

```
git checkout -b feature_x
```

vuelve a la rama principal

```
git checkout main
```

y borra la rama

```
git branch -d feature_x
```

Una rama nueva no estará disponible para los demás a menos que subas (push) la rama a tu repositorio remoto

```
git push origin <branch>
```

Actualiza & fusiona

Para actualizar tu repositorio local al commit más nuevo, ejecuta

```
git pull
```

en tu directorio de trabajo para bajar y fusionar los cambios remotos.

Para fusionar otra rama a tu rama activa (por ejemplo master), utiliza

```
git merge <branch>
```

en ambos casos git intentará fusionar automáticamente los cambios. Desafortunadamente, no siempre será posible y se podrán producir conflictos. Tú eres responsable de fusionar esos conflictos manualmente al editar los archivos mostrados por git. Después de modificarlos, necesitas marcarlos como fusionados con

```
git add <filename>
```

Antes de fusionar los cambios, puedes revisarlos usando

```
git diff <source_branch> <target_branch>
```

Log

Puedes obtener el histórico de commits con

git log

```
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700
```

```
    changed the version number
```

```
commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700
```

también puedes obtener una versión abreviada de esta información con

git log --pretty=oneline

```
ca82a6dff817ec66f44342007202690a93763949 changed the version number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test code
a11bef06a3f659402fe7563abf99ad00de2209e6 first commit
```

Reemplaza cambios

En caso de que hagas algo mal (lo que seguramente nunca suceda ;) puedes reemplazar cambios locales usando el comando

```
git checkout -- <filename>
```

Este comando reemplaza los cambios en tu directorio de trabajo con el último contenido de HEAD. Los cambios que ya han sido agregados al Index, así como también los nuevos archivos, se mantendrán sin cambio.

Por otro lado, si quieres deshacer todos los cambios locales y commits, puedes traer la última versión del servidor y apuntar a tu copia local principal de esta forma

```
git fetch origin  
git reset --hard origin/master
```

Si lo que ha ocurrido es que as agregado algún fichero al área de preparación y quieres quitarlo puedes hacer

```
git reset HEAD <filename>
```

Y para arreglar el último commit, para porejemplo agregar archivos al commit de los que te has olvidado

```
git commit -m 'initial commit'  
git add forgotten_file  
git commit --amend
```