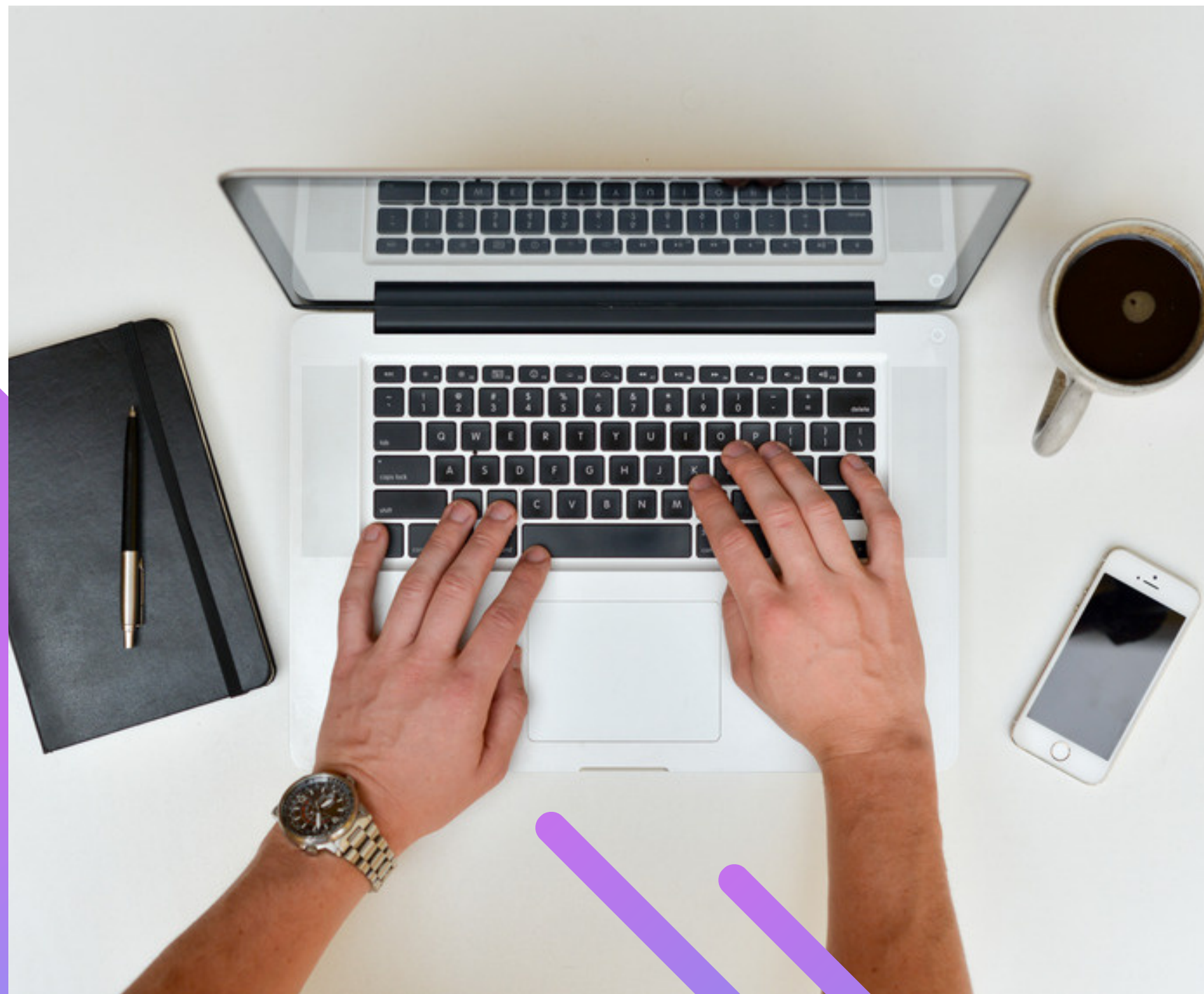




# CURSO GIT HUB

programacion software

GRUPO1



# Que es Git

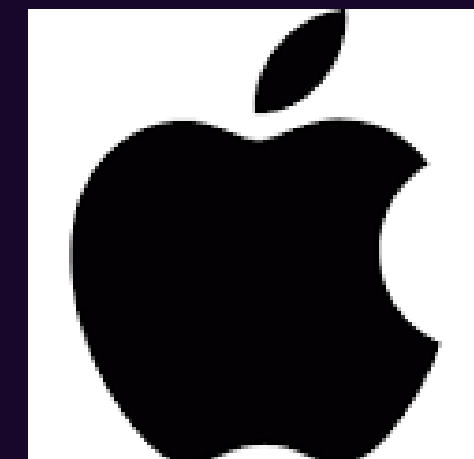
Git es una aplicacion de software libre para el desarrollo colaborativo proporcionando herramientas poderosas para el control de versiones, la colaboración y la gestión de proyectos de código abierto y privados.



# INSTALACION GIT

**Git es compatible con una gran variedad de sistemas operativos en el que podemos incluir a:**

- **Windows 7, 8, 8.1 y 10.**
- **macOS 10.9 (Mavericks) o posterior.**
- **En Linux Asegurarse de que tu distribución está actualizada y es compatible con la última versión de Git.**





# CONFIGURACION

Para la configuracion del git solo es necesaria realizarse la primera vez

Debemos abrir la terminal y escribir los siguientes comandos

Git config— global user.name "nombre del usuario" enter

Si se quiere cambiar el email

Git config— global user.email "correo" enter

Si queremos saber información de la configuración

Git config — list enter y aparecerá en pantalla

para abrir y editar Git config--global e

para salir de la configuración :q

```
-M5GBROU MINGW64 >>> Doc
repository in C:/Users/

-M5GBROU MINGW64 ~/Doc
er.name "Camila"

-M5GBROU MINGW64 ~/Doc
er.email "millanandane

-M5GBROU MINGW64 ~/Doc
ocal
rsion=0

rue

t14@outlook.com
```



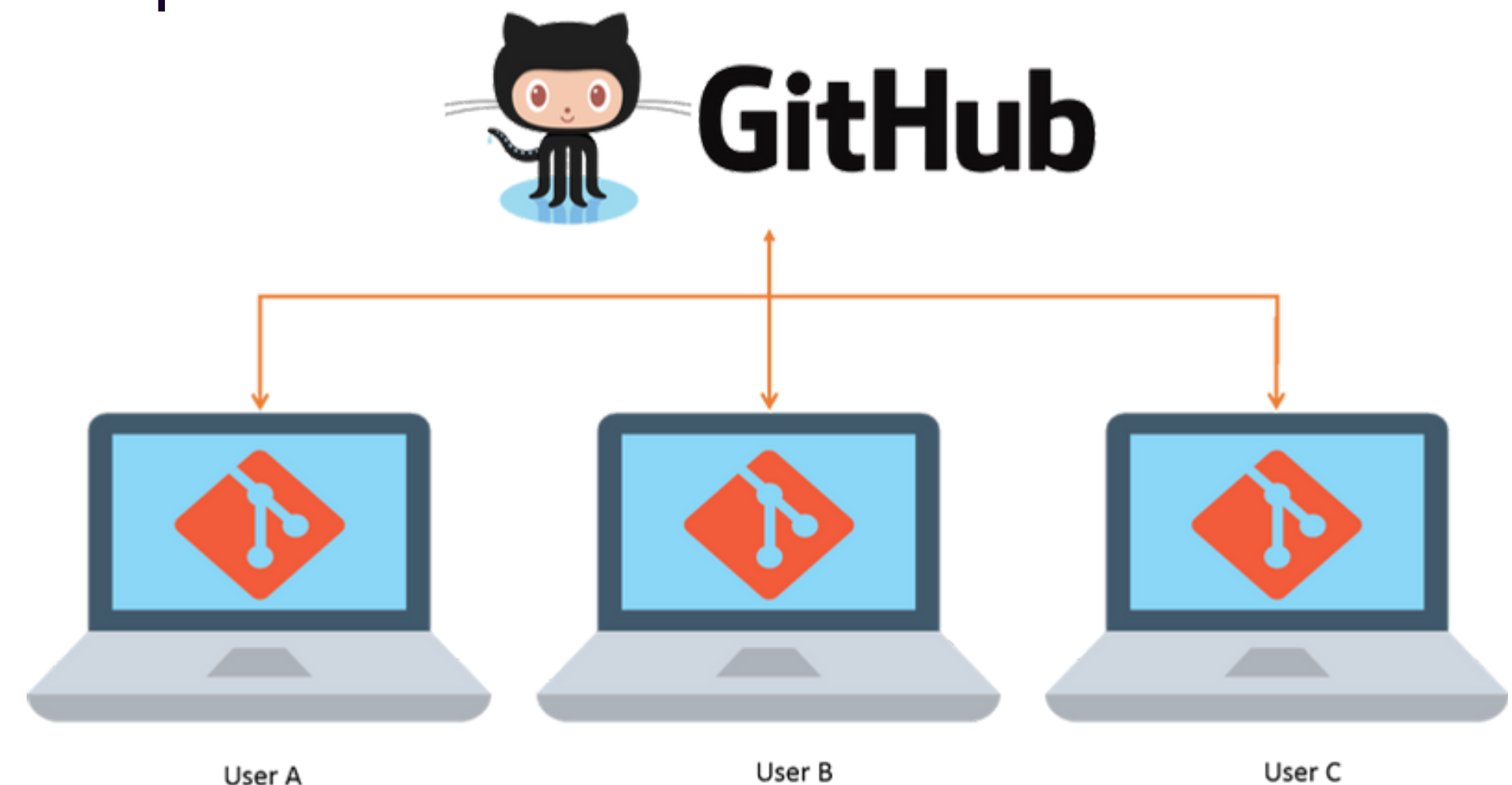
# CREAR REPOSITORIOS



En Git, un repositorio es un espacio donde se almacenan y se manejan los archivos de un proyecto, junto con un historial detallado de los cambios y las modificaciones realizadas en ellos a lo largo del tiempo, esto se lleva acabo por medio de los commits

Tipos de repositorios:

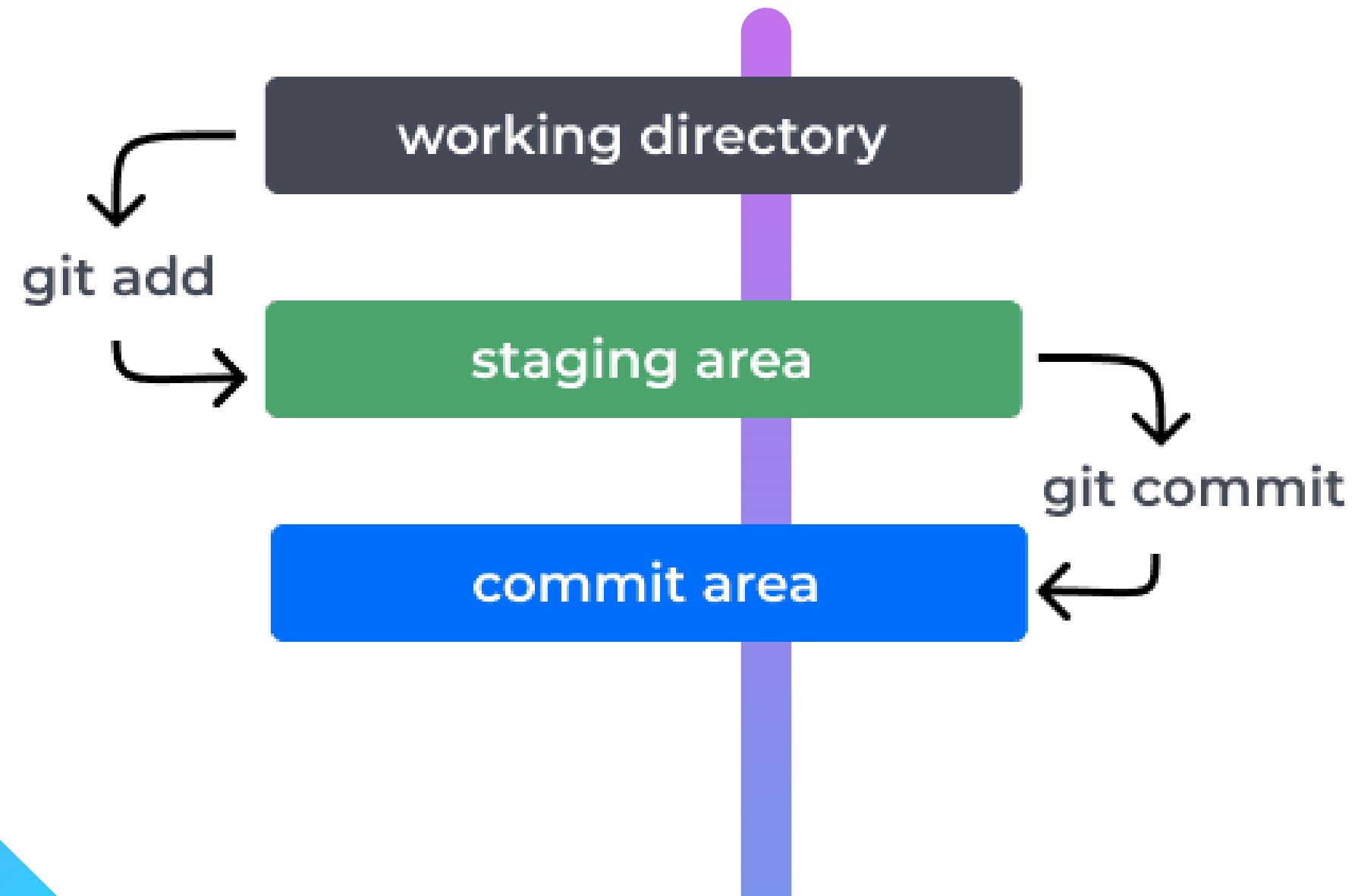
- Repositorio local: es una copia del repositorio en tu máquina local.
- Repositorio remoto: es una copia del repositorio que se encuentra en un servidor en línea.





# ESTADOS

tres estados:  
working directory,  
stading area,  
git directory (repository)





# REALIZAR FIRST COMMIT

Primero se hacen los cambios dentro del proyecto y luego de guardan los cambios

Para hacer el cambio se escribe  
subl. Nombre del archivo

Luego escribe git status enter

Git informa que existe un cambio

Git add ficheros a agregar

Git add . Agrega todos los cambios

Git status

Informa que están sin commit

Git commit -m seguido del mensaje

Crea un modo y los guarda

Git status aparece que todo está en orden

```
Test — -bash — 80x15
[apples-MacBook-Pro-4:test Khamosh$ git commit -m "first commit"
On branch master
nothing to commit, working tree clean
apples-MacBook-Pro-4:test Khamosh$
```

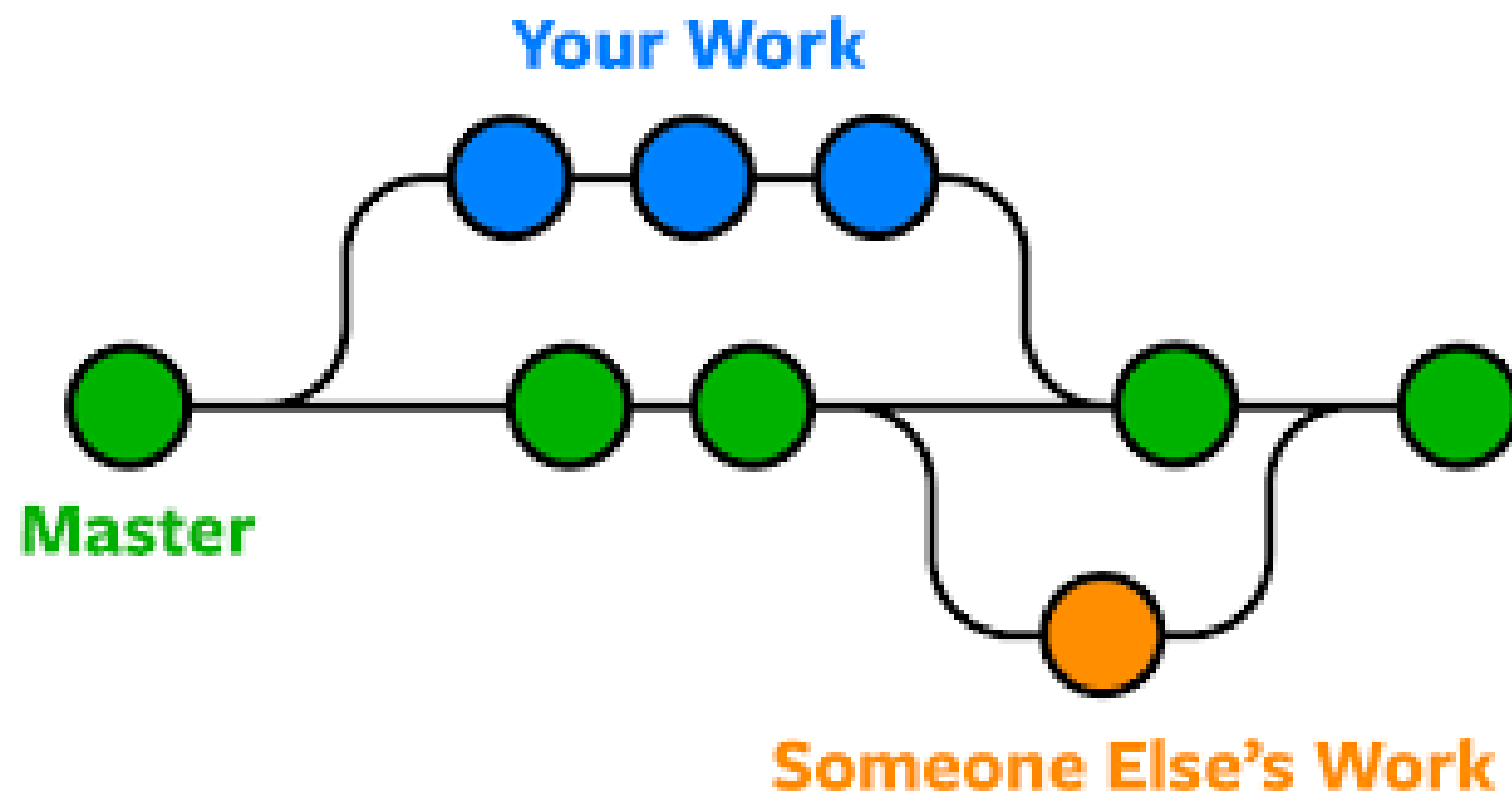




# CREAR RAMAS Y MOVER ENTRE ELLAS



En Git, puedes crear y moverte entre ramas para trabajar en diferentes líneas de desarrollo de tu proyecto.



01

Para crear una nueva rama, utiliza el comando `git branch` seguido del nombre de la nueva rama. Por ejemplo, si quieres crear una rama llamada "nueva-caracteristica", ejecuta:  
`git branch nueva-caracteristica`  
Ahora tienes una nueva rama llamada "nueva-caracteristica", pero aún no te has movido a ella.

02

Cambiar a una rama existente: `git checkout` seguido del nombre de la rama.  
`git checkout nueva-caracteristica`  
Otra forma más reciente de hacerlo es con el comando `git switch`:  
`git switch nueva-caracteristica`

03

Crear una nueva rama y cambiarte a ella en un solo paso:

Si prefieres crear una nueva rama y cambiarte a ella en un solo paso, puedes usar el comando `git checkout -b` (o `git switch -c` si estás utilizando Git 2.23 o superior). Por ejemplo:

Recuerda que es importante guardar tus cambios utilizando `git commit` antes de cambiar de rama, ya que los cambios no confirmados no se trasladarán automáticamente a la nueva rama. Cambiar de rama descartará los cambios no confirmados en la rama actual.

04

Para fusionar dos ramas en Git, generalmente se utiliza el comando `git merge`. Aquí te explico los pasos básicos para fusionar dos ramas:  
`git checkout main`  
Esto cambia a la rama "main" como tu rama actual.





# Fin

