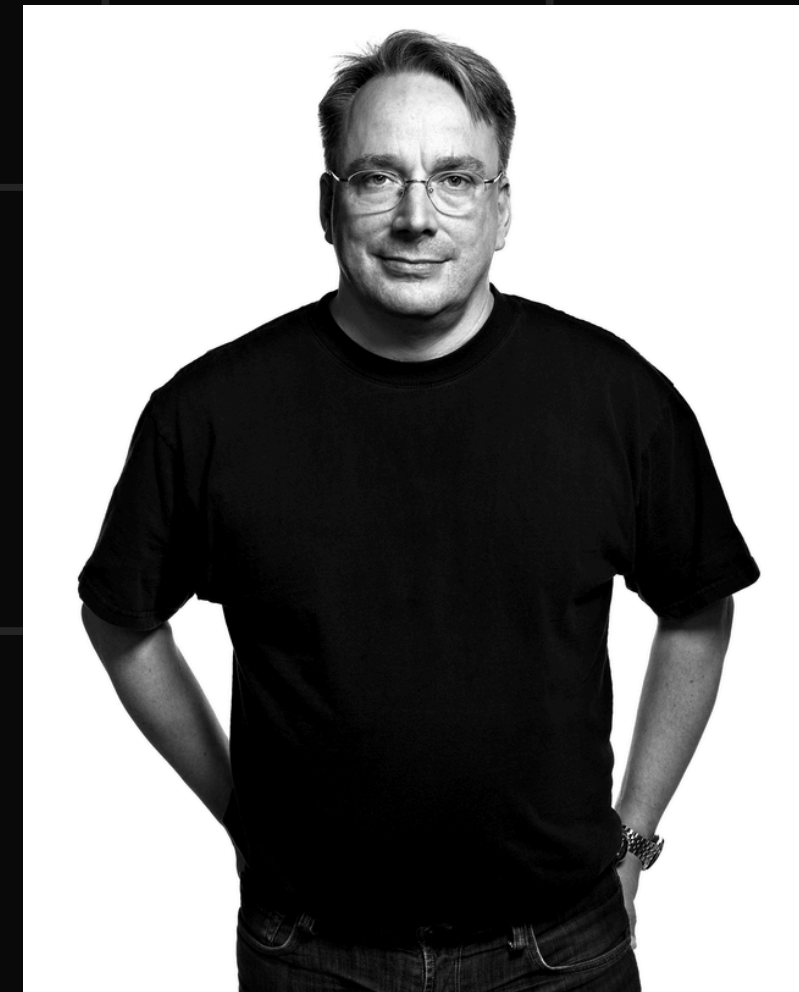


# GIT

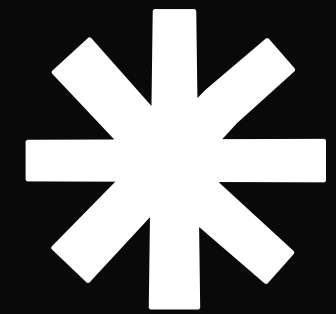


# \*\*\* Introducción

Git es un sistema de control de versiones distribuido que permite a varios desarrolladores trabajar simultáneamente en un proyecto, rastreando y gestionando cambios en el código fuente. Fue creado por Linus Torvalds en 2005 para el desarrollo del kernel de Linux, y desde entonces se ha convertido en una herramienta esencial en el desarrollo de software.



# Conceptos Básicos de Git



## Repositorio

La estructura donde Git almacena el código fuente y el historial completo de todos los cambios realizados en un proyecto. Un repositorio puede estar localizado en tu máquina (local) o en un servidor (remoto), como GitHub, GitLab, o Bitbucket.

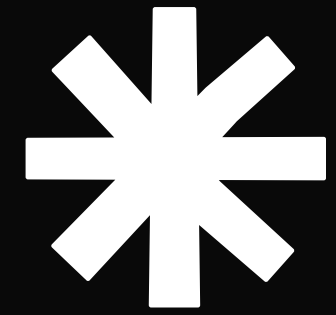
## Merging

El proceso de combinar los cambios de una rama en otra. Generalmente, se utiliza para integrar los cambios de una rama de característica en la rama principal. Durante la fusión, Git intentará combinar automáticamente los cambios; sin embargo, si hay cambios conflictivos, se producirá un conflicto que deberá ser resuelto manualmente.

## Branch

Una línea de desarrollo independiente dentro de un repositorio. Las ramas permiten a los desarrolladores trabajar en diferentes características o soluciones a problemas de manera aislada, sin afectar el código en la rama principal.

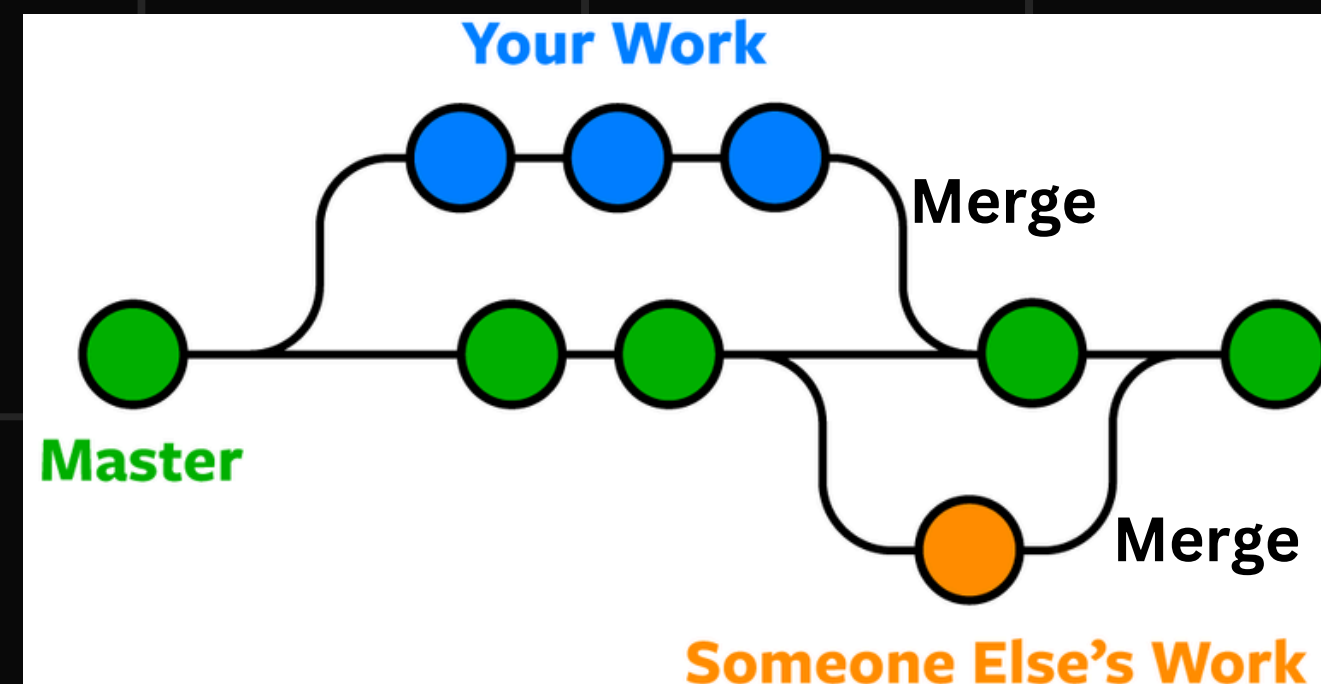
# Conceptos Básicos de Git



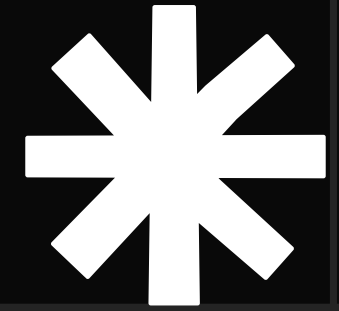
## Branch

Una rama en Git es una línea independiente de desarrollo dentro de un repositorio. Las ramas permiten trabajar en diferentes características, correcciones o experimentos sin afectar la rama principal (generalmente llamada main o master).

- Rama Principal (main o master): Es la rama por defecto donde se mantiene el código de producción o la versión principal del proyecto.
- Rama de Característica: Las ramas se suelen crear para desarrollar nuevas características, corregir errores o realizar experimentos sin afectar el código principal.

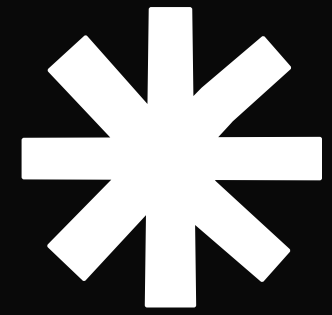


# Comandos de las branches



- Muestra una lista de todas las ramas locales en el repositorio.
  - `git branch`
- Muestra una lista de todas las ramas remotas.
  - `git branch -r`
- Muestra una lista de todas las ramas remotas y locales
  - `git branch -a`
- Eliminar una rama
  - `git branch -d`
- Cambiarte de rama
  - `git checkout`
- Crear una nueva rama
  - `git checkout -b`

# Conceptos Básicos de Git



## Commit

Es una "instantánea" del estado actual del proyecto. Cada vez que haces un commit, Git guarda el estado de todos los archivos en el proyecto en ese momento, creando un punto de referencia que puedes volver a consultar más tarde.

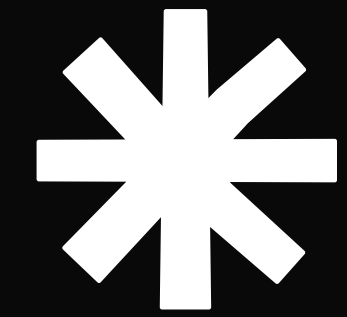
## Staging Area

Área intermedia donde Git prepara los cambios que se incluirán en el próximo commit. Cuando haces un cambio en tu código, primero lo añades al área de staging, y luego puedes hacer un commit de todos los cambios añadidos a esta área. Se usa el comando “add”.

## Cloning

El proceso de crear una copia completa de un repositorio remoto en tu máquina local. Cuando clonas un repositorio, obtienes todo el historial del proyecto y todas sus ramas.

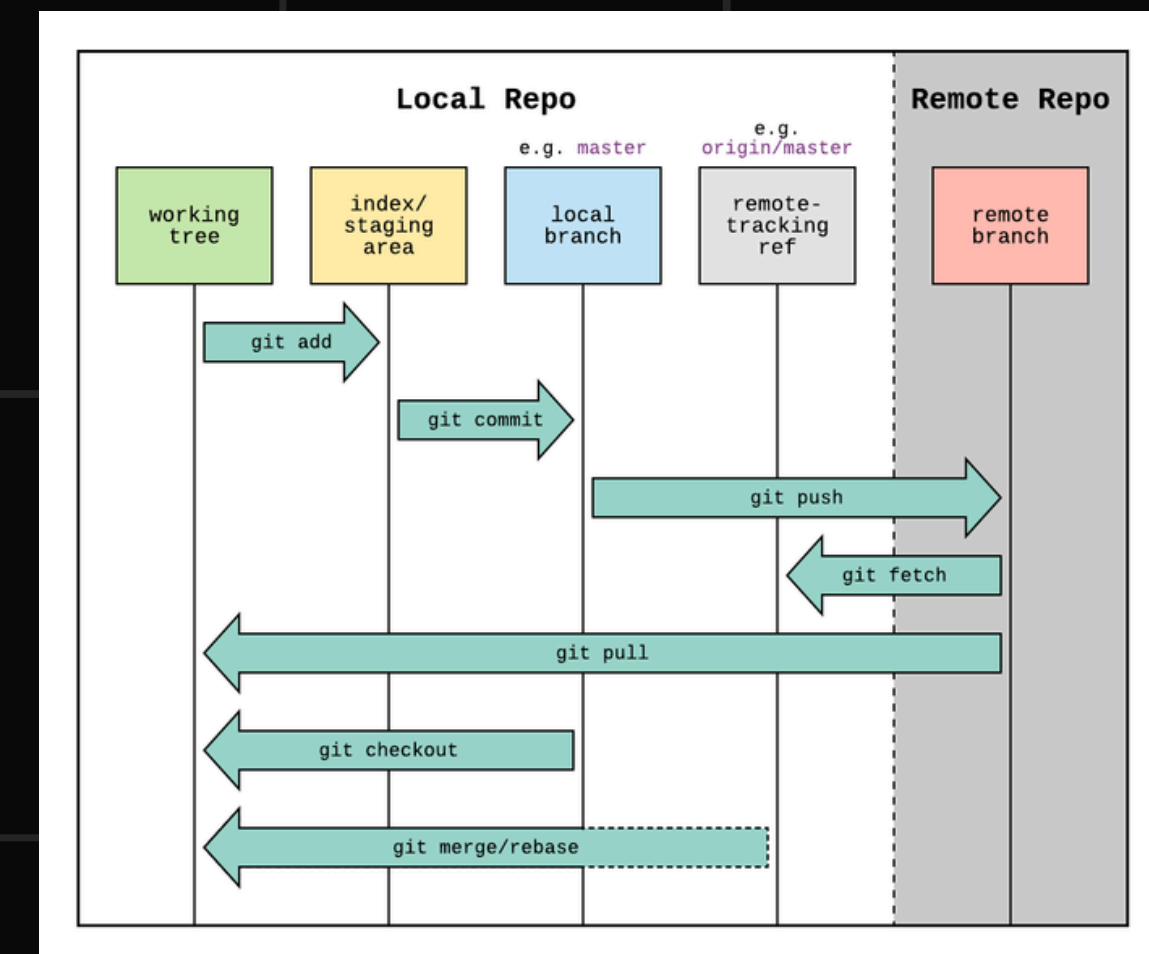
# Conceptos Básicos de Git



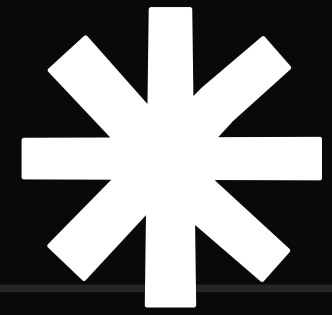
## Remote

Es una versión del repositorio que está alojada en un servidor o en la nube, como en GitHub, GitLab o Bitbucket. Puedes interactuar con remotos para compartir tus cambios con otros desarrolladores o para obtener actualizaciones de ellos.

- git push: Este comando se usa para enviar los commits locales a un repositorio remoto.
- git pull: Este comando se usa para traer los cambios de un repositorio remoto y combinarlos con tu rama local.
- git fetch: Este comando descarga la información de la rama desde el repositorio remoto, pero no combina los cambios con tu rama local.



# Configuración inicial

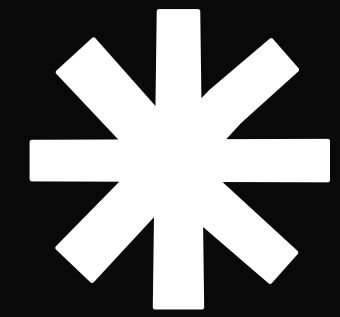


Antes de empezar a trabajar con Git, es importante configurar tu nombre de usuario y tu correo electrónico. Esto asegura que tus commits sean atribuidos correctamente a ti.

- Configurar Nombre de Usuario:
  - `git config --global user.name "Tu Nombre"`
- Configurar Correo Electrónico:
  - `git config --global user.email "tu.email@ejemplo.com"`
- Verificar Configuración:
  - `git config --list`



# Flujo de trabajo en git



## 1. Crear o clonar el repositorio

- `git init`
- `git clone https://github.com/usuario/repositorio.git`

## 2. Crear una rama de característica

Crea una nueva rama para trabajar en una característica o corrección específica. Esto mantiene el desarrollo aislado de la rama principal.

- `git checkout -b nombre_de_rama`

## 3. Realizar Cambios y Hacer Commits

Trabaja en la nueva característica o corrección en tu rama, añade los cambios al área de staging y haz commits para registrar el progreso.

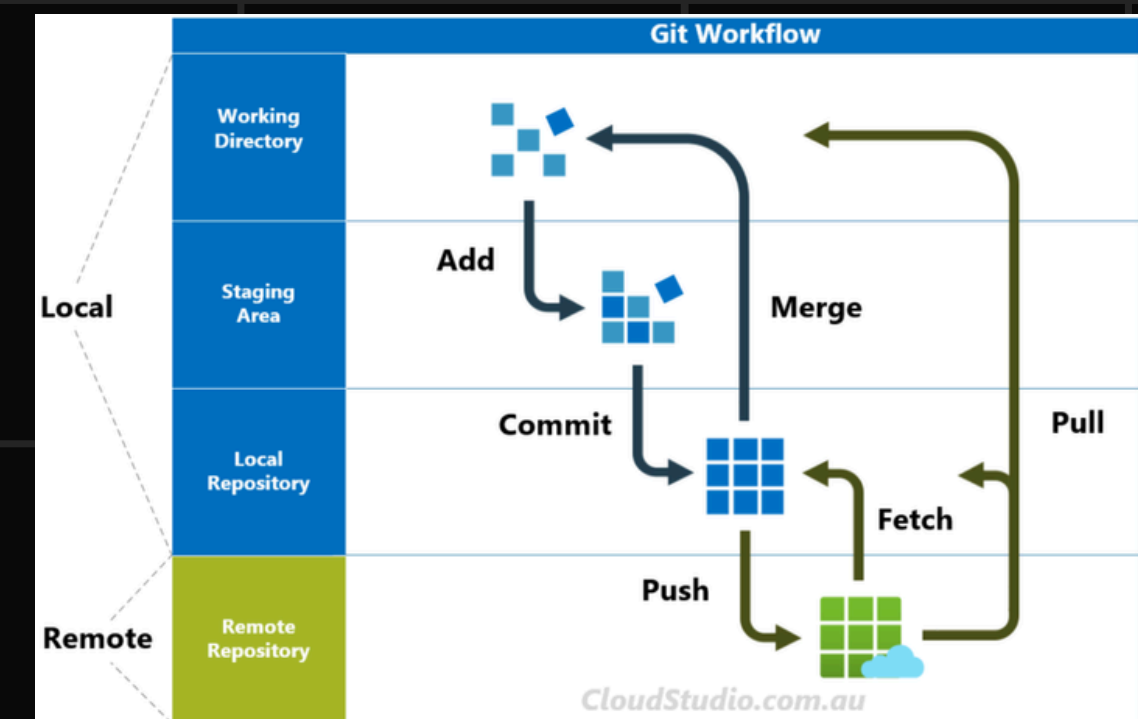
- `git add archivo_modificado`
- `git commit -m "Descripción de los cambios"`

## 4. Fusionar la Rama de Característica

- `git checkout main`
- `git merge nombre_de_rama`

## 5. Subir los Cambios al Repositorio Remoto

- `git push origin main`



# Comandos para el Repositorio \*

## Ver el Estado del Repositorio

- Ver el Estado Actual: Muestra los archivos modificados, los archivos que están en el área de staging, y los archivos que no están rastreados.
  - `git status`

## Ver el Historial de Commits

- Mostrar el Historial de Commits: Muestra una lista de commits en el repositorio, incluyendo el hash del commit, el autor, la fecha y el mensaje del commit.
  - `git log`