

**MAKE**  **REAL**  
**CAMP**

- **Listando carpetas y archivos:**  
-> `ls`
- **Cambiando de ubicación:**  
-> `cd /`
- **Creando una carpeta:**  
-> `mkdir ejercicios`
- **Eliminando archivos y carpetas:**  
-> `rm -rf node_modules`
- **Moviendo archivos y carpetas:**  
-> `mv archivo1.txt mi_carpeta`
- **Copiando archivos y carpetas:**  
-> `cp archivo1.txt mi_carpeta`



A screenshot of a macOS terminal window. The title bar shows the user 'khriztianmoreno@Osiris' and the current directory '~/dev'. The terminal displays the following commands and output:

```
>  
➤ ~/dev ..... 09:21:53 AM  
> ls  
➤ honest ➤ makeitreal ➤ mattersupply ➤ personal  
➤ ~/dev ..... 09:22:03 AM  
> |
```

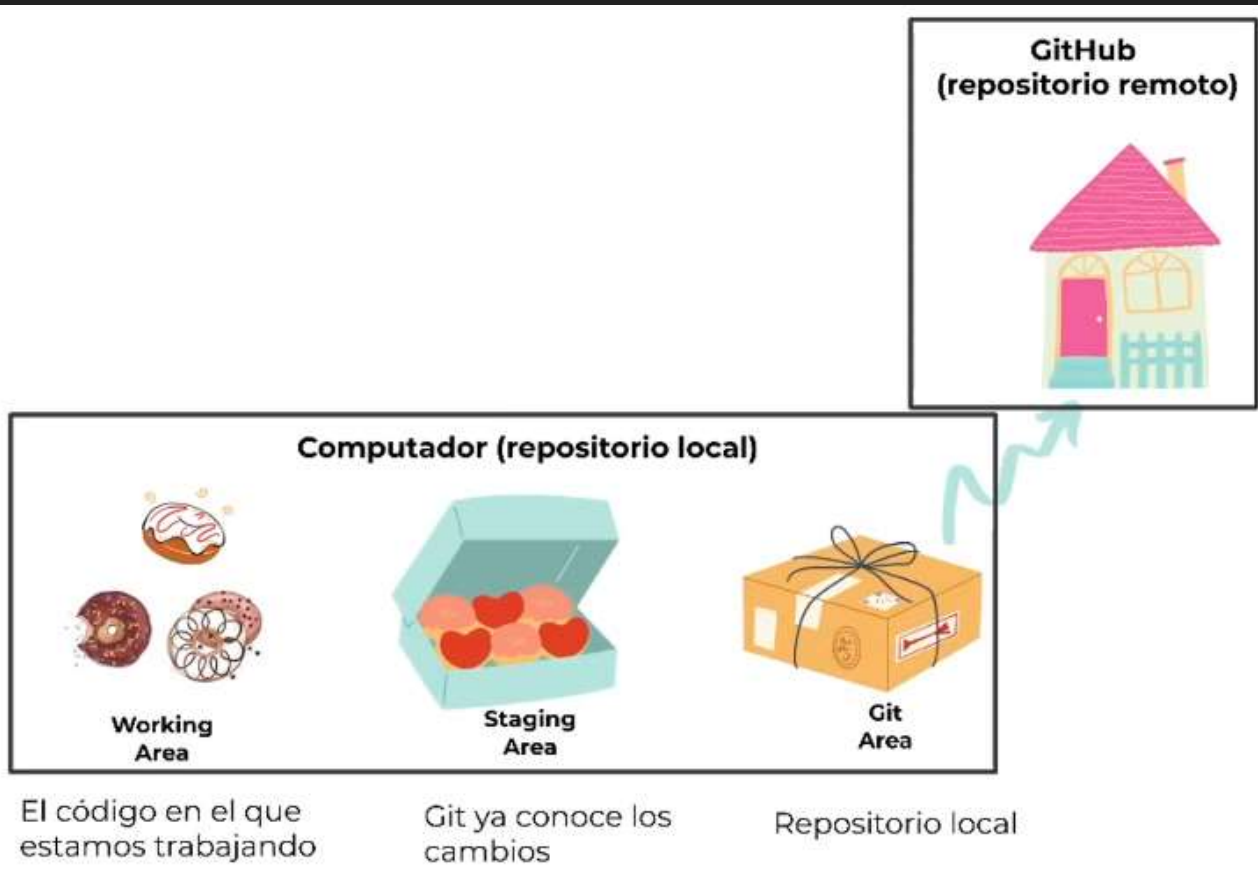
# Git y Github



- Instalación
- Configuración
- Ramas o Branches
- Repositorios: local y remoto

1. **git init** : Este comando crea un nuevo repositorio de Git vacío en el directorio actual. Es el primer paso para comenzar a utilizar Git.
2. **git add** : Este comando agrega archivos al área de preparación (también conocida como "staging area"), que es donde se almacenan los cambios que se van a incluir en el próximo commit.
3. **git commit** : Este comando crea un nuevo commit, que es una instantánea del estado actual del repositorio. Los cambios en el área de preparación se incluyen en el commit.
4. **git status** : Este comando muestra el estado actual del repositorio, incluyendo los archivos modificados, los archivos nuevos y los cambios que están en el área de preparación.
5. **git log** : Este comando muestra el historial de commits del repositorio, incluyendo el autor, la fecha y la hora del commit, y el mensaje del commit.
6. **git checkout** : Este comando permite cambiar entre ramas o restaurar archivos a una versión anterior. Puede ser utilizado para cambiar entre ramas o para crear una nueva.
7. **git clone** : Este comando crea una copia completa de un repositorio en otro directorio. Es útil para trabajar en un proyecto en una computadora diferente o para colaborar con otros desarrolladores.
8. **git push** : Este comando envía los cambios locales al repositorio remoto. Es importante enviar cambios al repositorio remoto para mantener una copia de seguridad y compartir cambios con otros desarrolladores.
9. **git pull** : Este comando descarga los cambios más recientes del repositorio remoto y los fusiona con la copia local. Es importante actualizar la copia local con los cambios más recientes del repositorio remoto.

# Áreas de Git





Estamos Trabajando en  
la misión 2

Git es una herramienta poderosa para manejar versiones de código: Con Git, es fácil mantener diferentes versiones de tu código, crear y fusionar ramas para trabajar en nuevas características sin afectar la rama principal.

## Git

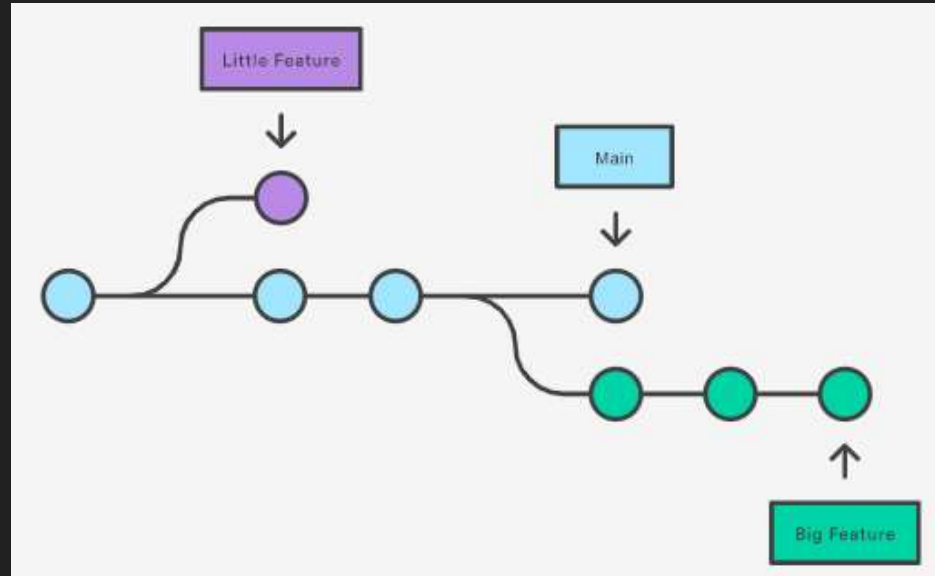
- Sistema de control de versiones
- De código abierto
- Se utiliza a través de la línea de comandos

## GitHub

- Empresa que ofrece hosting para proyectos de software
- Tiene interfaz gráfica en la web
- Provee herramientas para el trabajo colaborativo

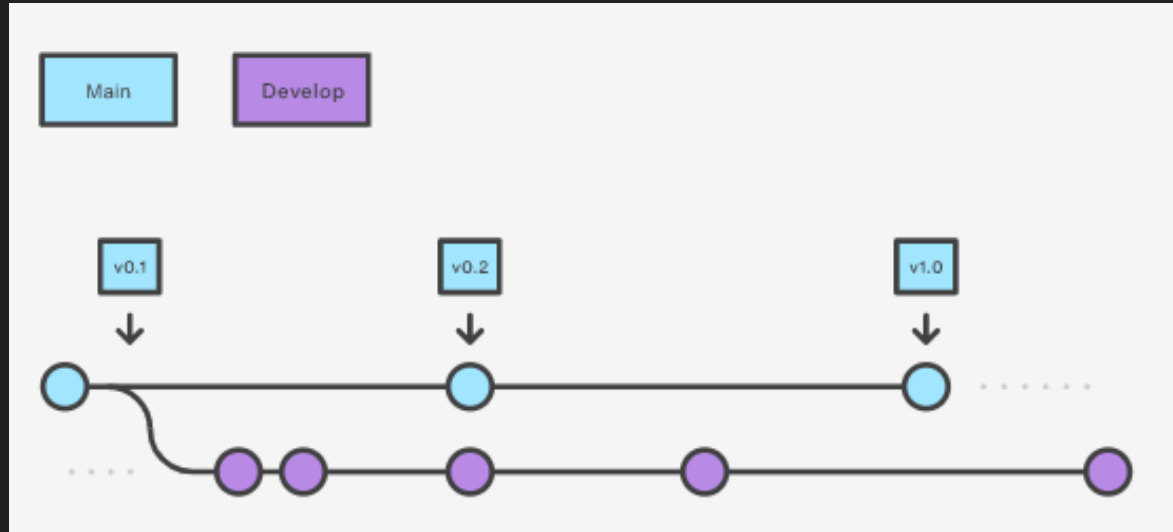


## Git Branches o Ramas



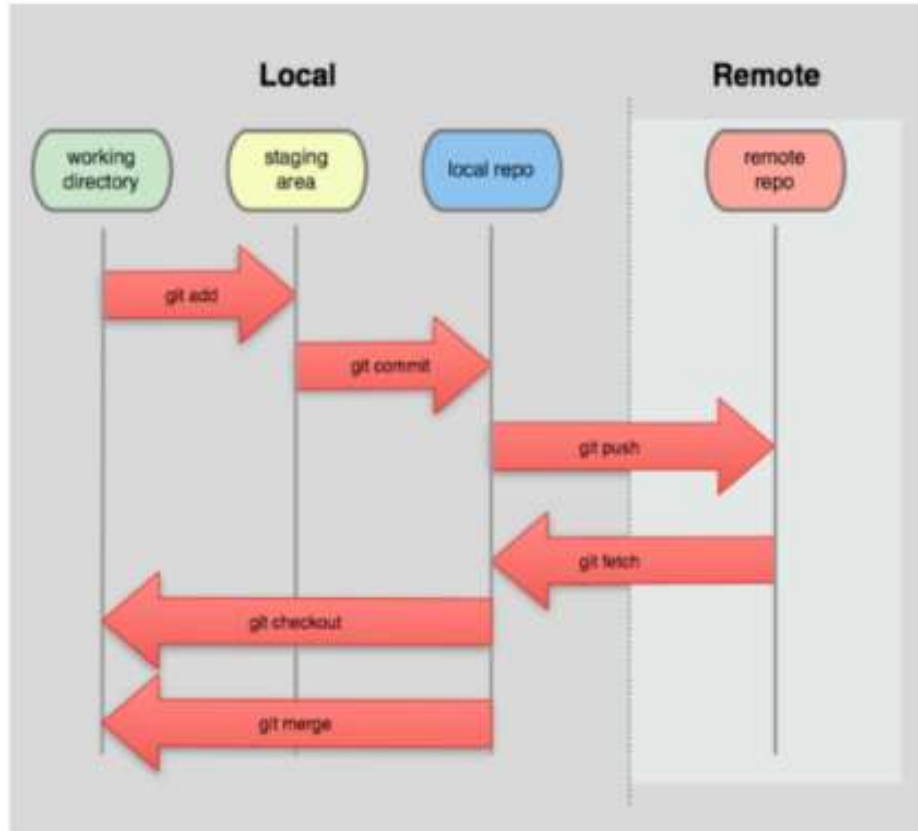
Las ramas son punteros que guardan los cambios realizados por el desarrollador.

# Gitflow



Modelo de Git Branching que involucra la utilización de ramas con propósitos específicos.

# Flujo de trabajo con Git





# Git workflow diagram





Estamos Trabajando en  
la misión 3

## Tener en cuenta

Comandos de git.

Creación de ramas de git.

Creación y clonación de repositorio.

Control de versionamiento.

Notación Markdown.

## Próximas Temáticas

HTML: etiquetas y estructura.

CSS: manejo de estilos para vistas.