

# Introducción a Git y Github



# Version Control System

- Forma de manejar archivos y directorios
- Monitorear cambios a través del tiempo
- Ver versiones anteriores

# Git

- Creado por Linus Torvalds, Abril 2005
- Creado para manejar los cambios en el kernel de Linux
- Un programa controlador de versiones de consola
- Usa la suma de verificación para garantizar la integridad de los datos
- Distribuido (Sin servidor, desarrollador es el cliente, servidor y repositorio )
- Multiplataforma
- OpenSource



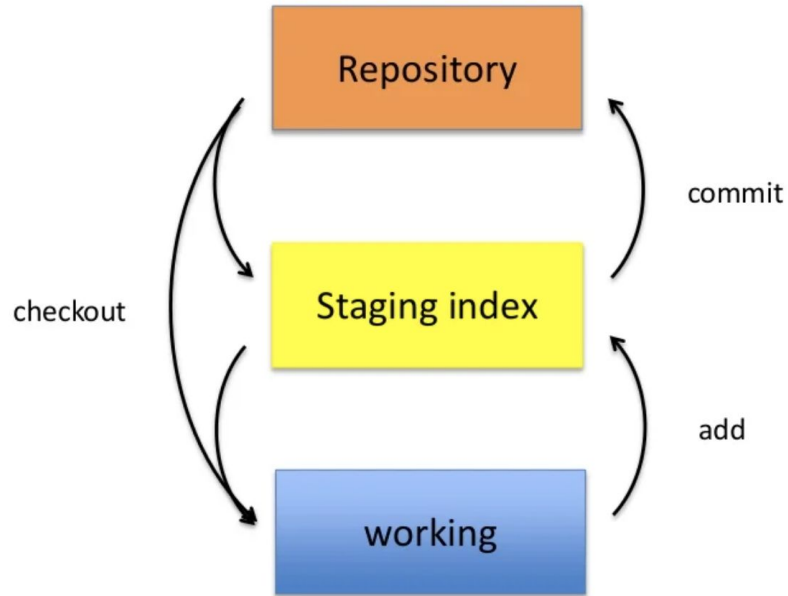
# Control de versiones distribuido

- No hay necesidad de un servidor central
- Funciona sin internet
- No hay punto único de fallo
- Los devs pueden trabajar de forma independiente y juntar su código después
- Toda copia de git tiene la historia completa del repo
- Git guarda cambios no versiones

# Repositorio

- “Repo” aka Repositorio
- Organiza un único proyecto
- Contiene carpetas, imágenes, archivos, videos, etc

# Arquitectura de git

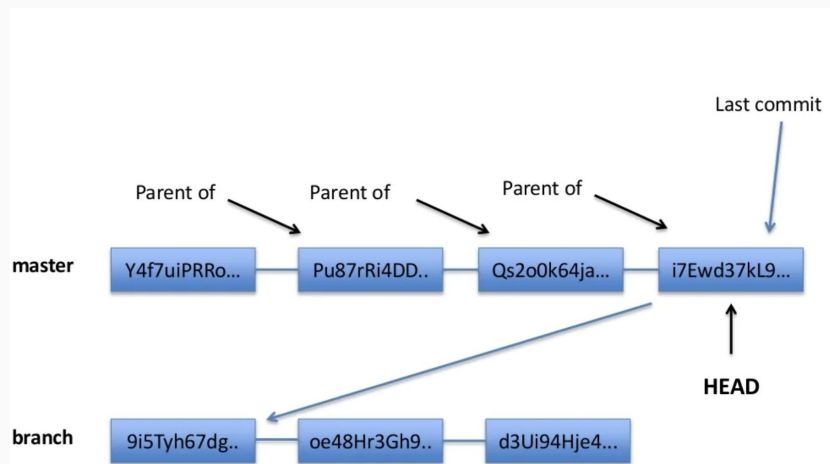


# Comandos

- `git init`
- `git add <file>`
- `git commit -m <msg>`
- `git log`
- `git status`
- `git diff`
- `git checkout`
- `git reset`

# HEAD

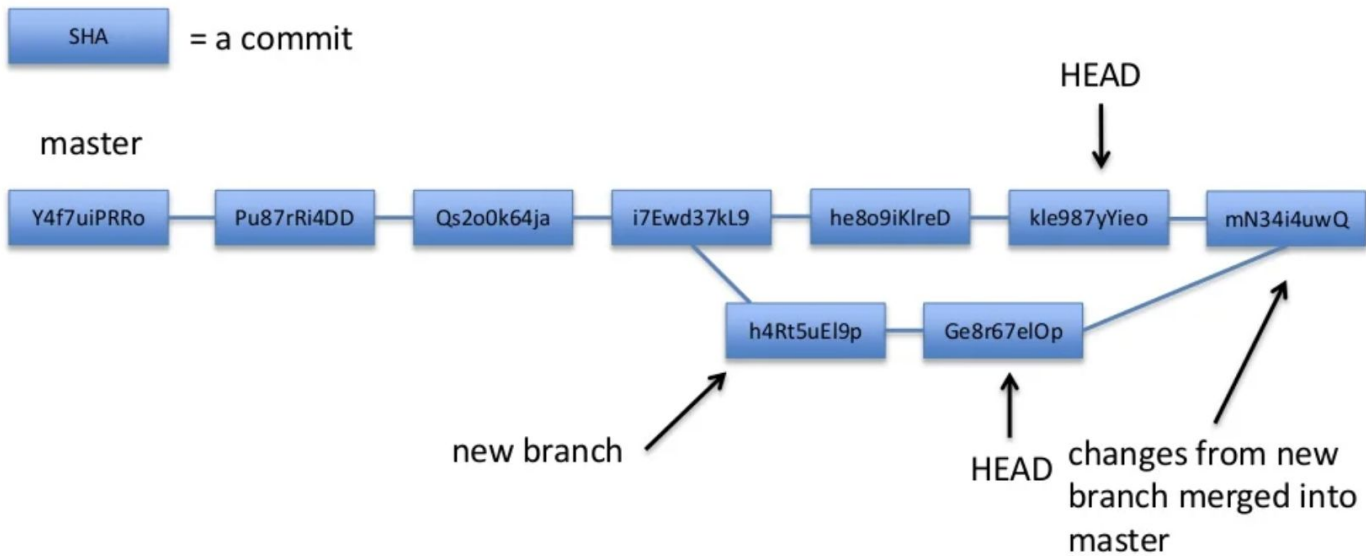
- Puntero a un commit específico en el repositorio
- Apunta al padre del proximo commit





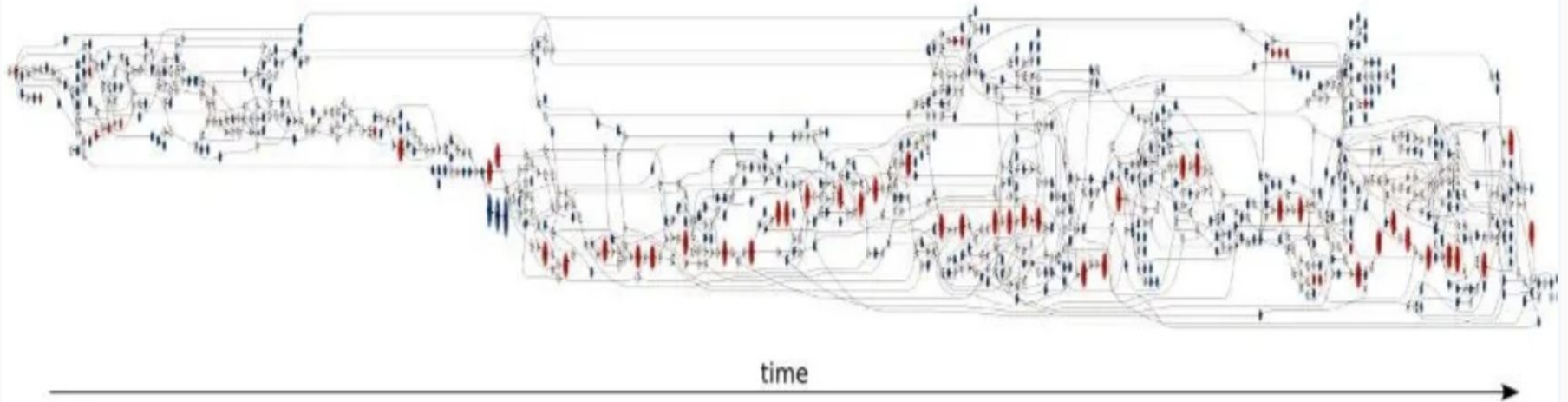
# “Branching”

- Ayuda a experimentar sin miedo a romper el código
- Si algo no funciona no hay que deshacer muchos cambios a la rama principal
- Si funciona se puede “mergear” a master
- Solo hay un directorio



# Visualización de un repo

16 forks and 7 contributors to the master branch



# “Merge Conflicts”

Cuando hay cambios en la misma línea en diferentes commits se genera un “merge conflict”.

Para resolverlo se puede:

- Abortar el merge “git merge -abort”
- Manualmente resolver el conflicto
- Usar una herramienta para merge

# Github

- Almacena repositorios de git
- Plataforma de almacenamiento de repos mas usada

