

# Linux Básico

## Cuarta Clase



# **Administración del sistema (sysadmin)**

# K.I.S.S. principle

- Prioriza la simplicidad (*"Simplicidad es la última sofisticación"*)
  - No sobre pienses soluciones
  - Probablemente alguien que sabe más que tu ya lo hizo y documentó
- Documenta todo lo que hagas
  - En 3 meses no recordarás los comandos exactos que usaste
  - No sabes si el blog con el que resolviste el problema seguirá ahí
- Mantente lo mas cercano a lo default que te sea posible y conveniente
  - Lo agradecerás al reinstalar
  - La ayuda externa se dificulta mientras más personalizada sea tu instalación
- Reduce riesgos
  - Cada configuración o programa puede tener una vulnerabilidad, mientras más común sea más fácil será detectarla

# Introduccion a git

# Git y el control de versiones

## Control de versiones

- Mantienen registros de los cambios
- Permite desarrollo colaborativo
- Registra quien hace los cambios y cuando
- Permite revertir cambios

## Git

- Usuarios tienen toda la historia del código en su máquina
- Puede manipularse sin internet
- Los cambios públicos se generan hasta que se manda al servidor
- 2005 por Linux Torvalds

# Conceptos básicos

## Snapshot

- Registro de la historia del código
- El código en un punto del tiempo exacto
- Tu decides cuando
- Puedes regresar a visitarlas
- Se quedarán por la eternidad

## Commit

- La acción de crear la Snapshots
- Al final un proyecto es el conjunto de muchos commits
- Contiene
  - Información de cómo los archivos cambiaron
  - Referencia al commit anterior (parent)
  - Un hash de nombre

# Conceptos básicos

## Repositorio (“repo”)

- La colección de todo tu trabajo, un bonche de commmits
- Puede vivir en tu máquina, o en una máquina remota (Gitlab/Github)
- El acto de copiar repositorios es el acto de clonar
- El acto de jalar cambios con los que no cuentas es ‘pull’
- El acto de enviar tus cambios es un ‘push’

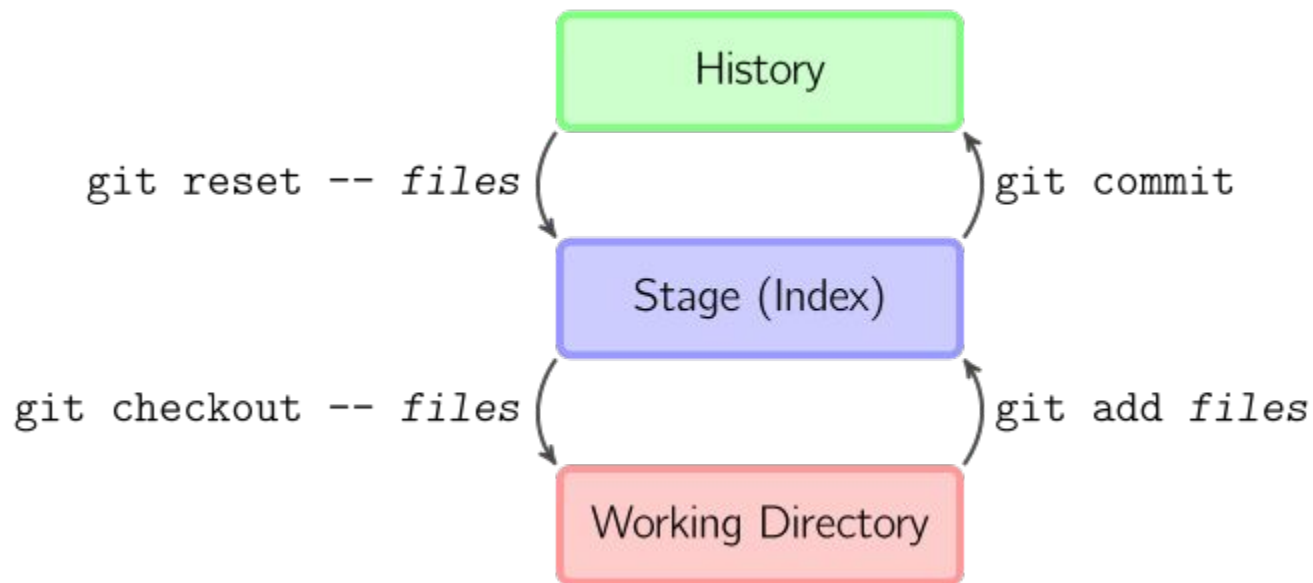
## Branches y proyectos

- Rama
  - Una colección de commits en orden
  - La default es main, todo es una rama
  - Puede haber millones
- Proyecto
  - Suelen ser un bonche de commits en una rama que vive en algún repositorio.

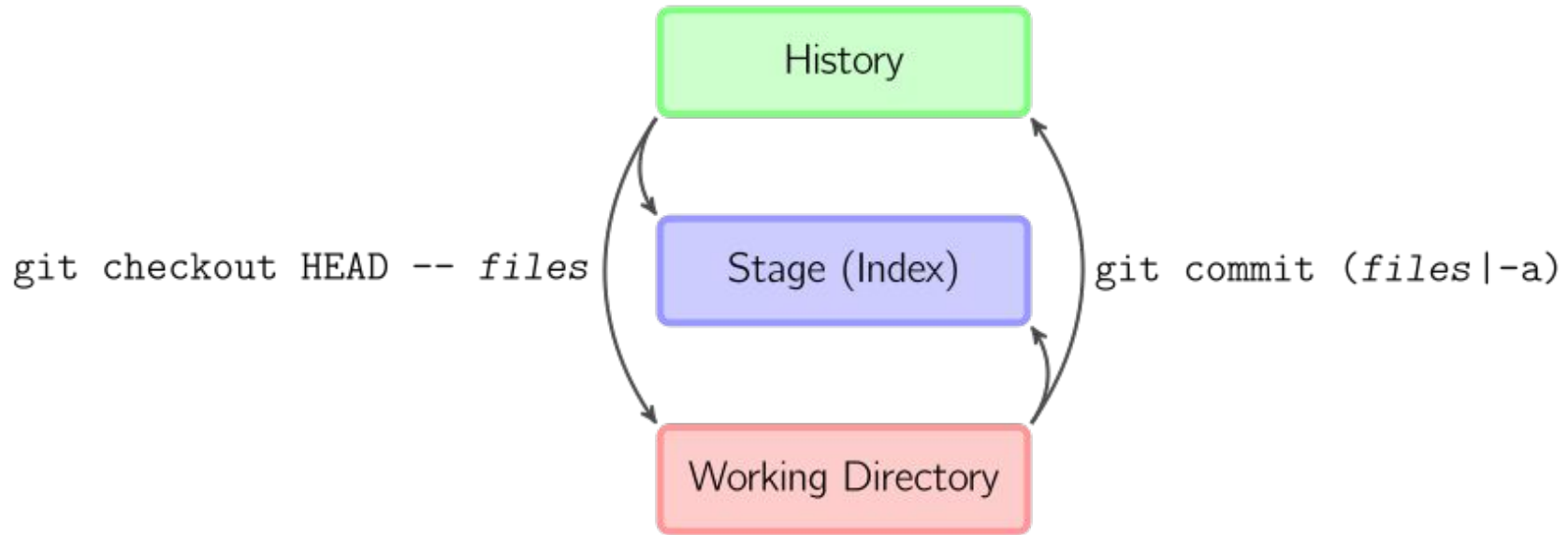
# Uso de git



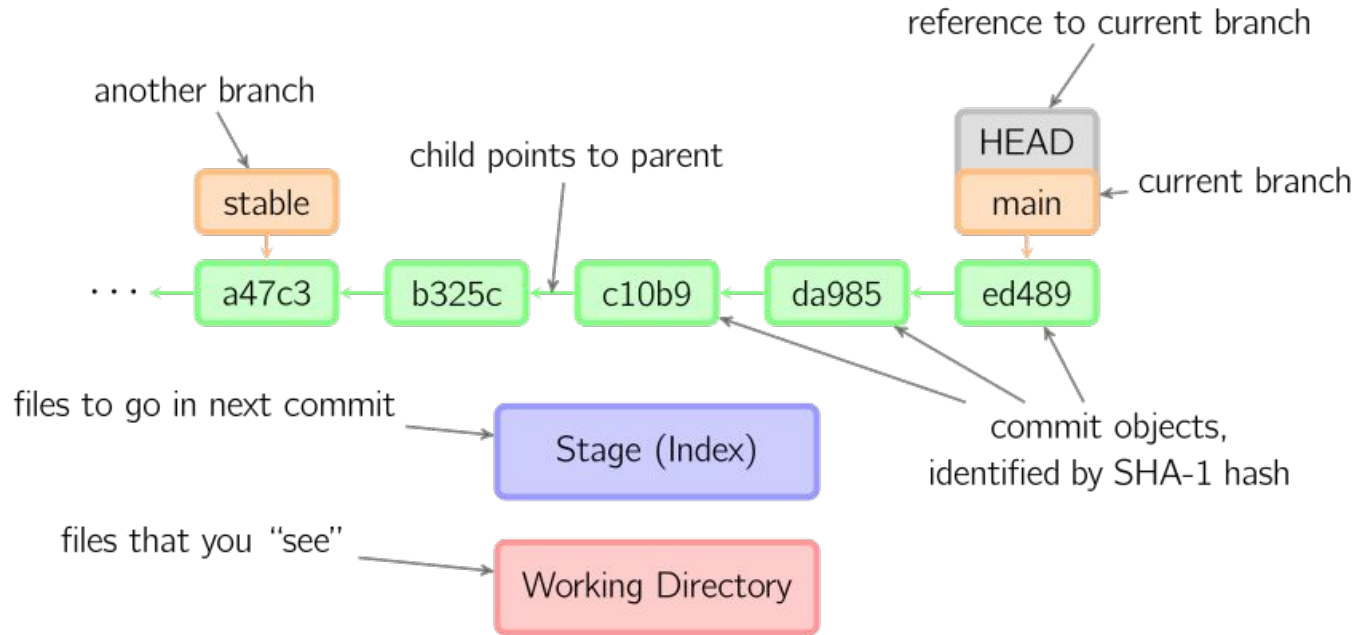
# Estructura



# Atajos

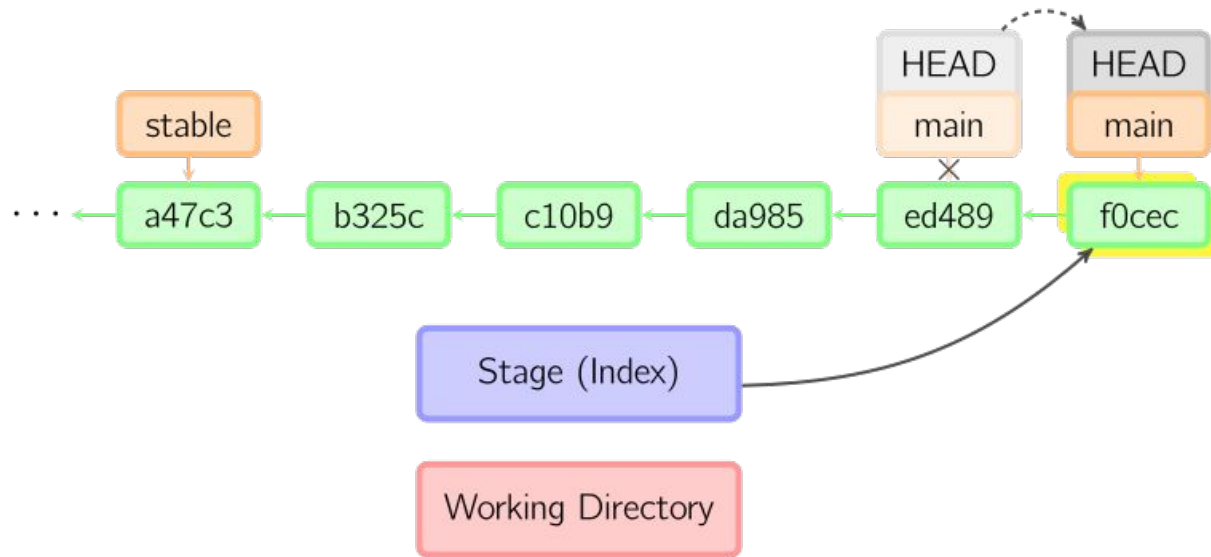


# Estructura completa



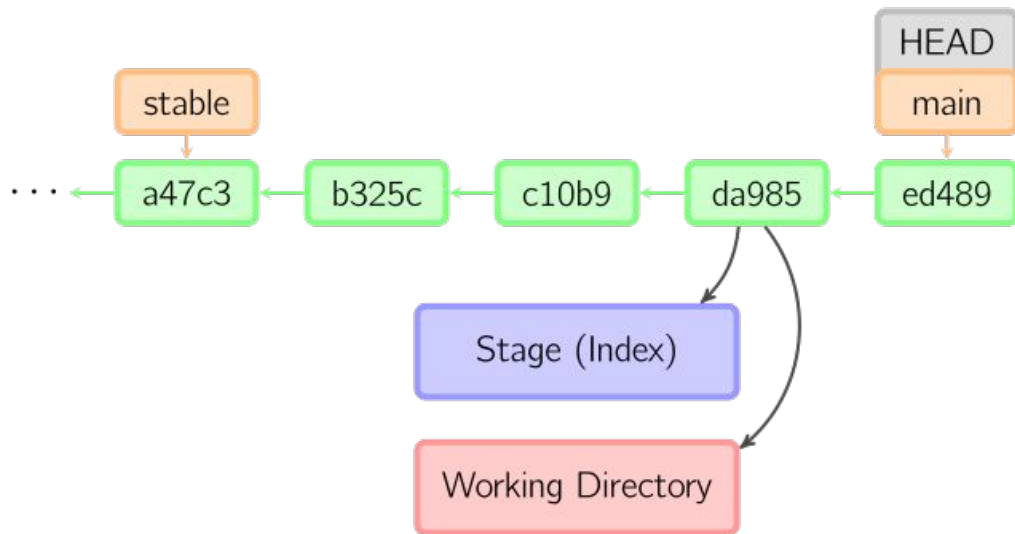
# Commit

`git commit`



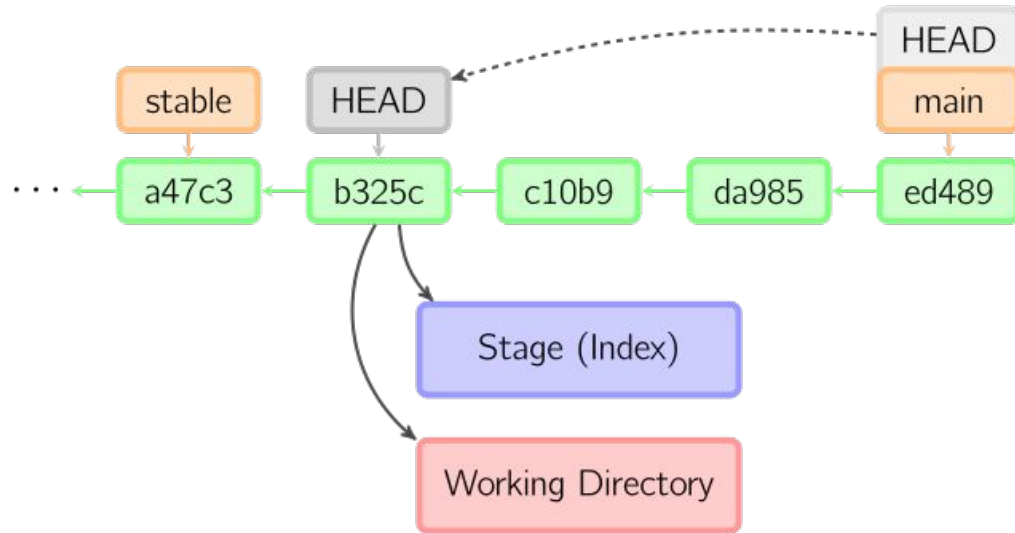
# Retroceder un commit

```
git checkout HEAD~ files
```

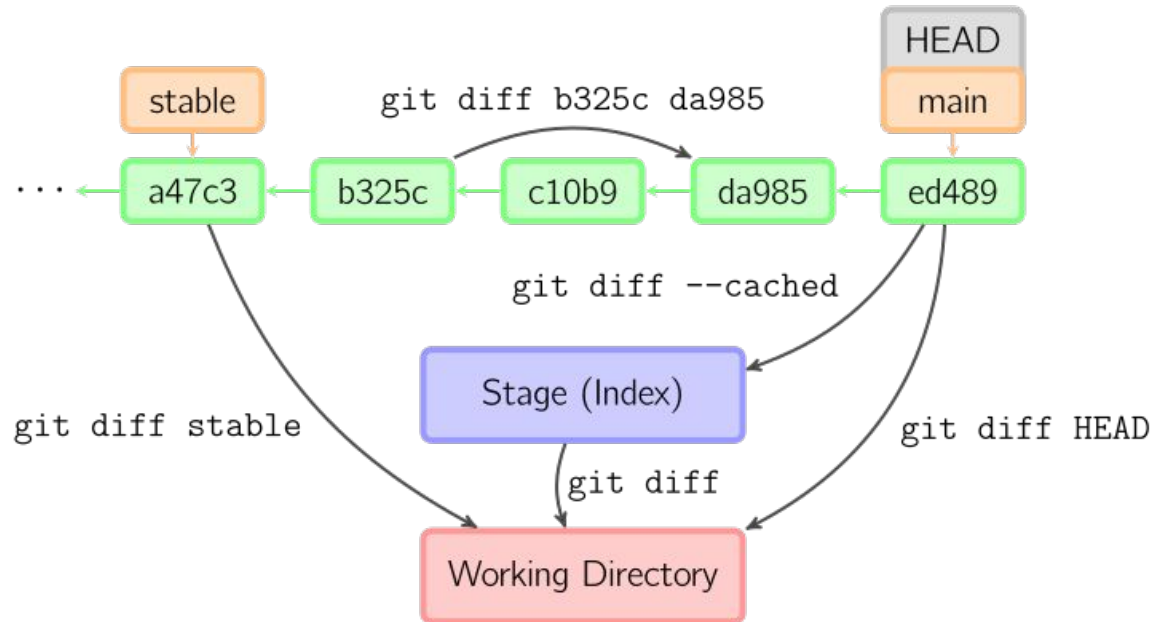


# Retroceder n commits

`git checkout main~3`

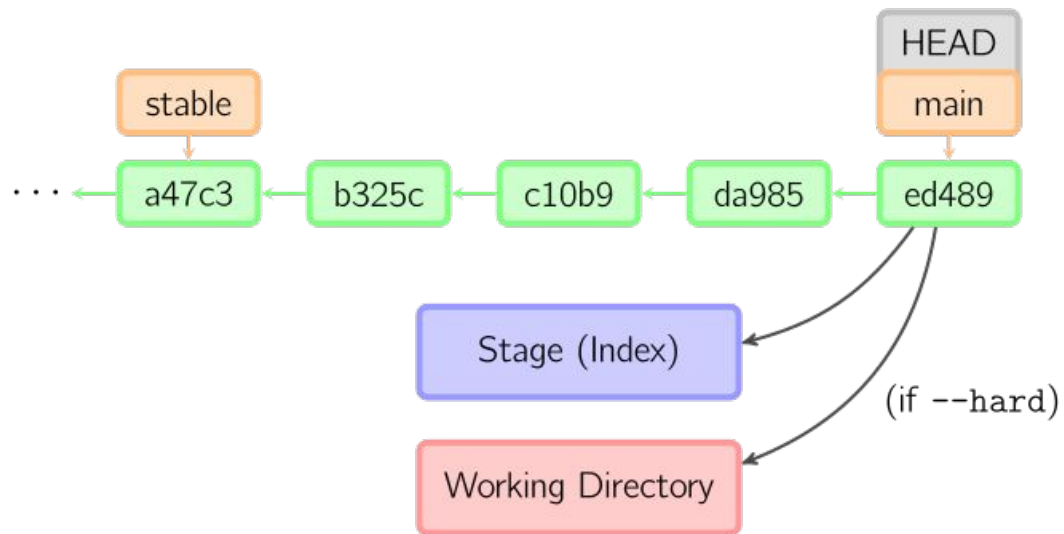


# Diferencias



# Reset

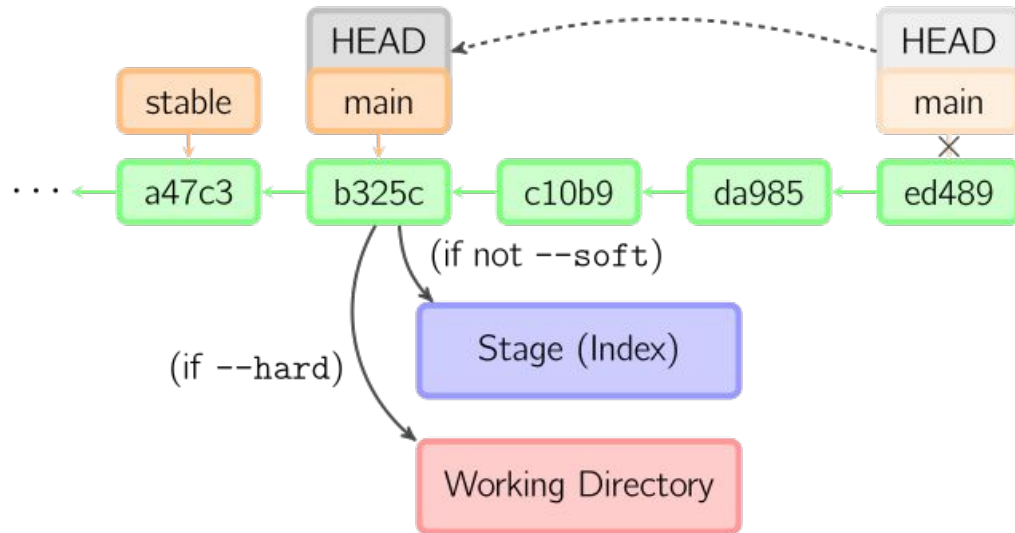
`git reset`





# Reset a commit anterior

`git reset HEAD~3`



# Resumen de comandos

- `git init`: Inicial repositorio
- `git add <archivo>`: Agregar archivos a stage
- `git commit`: Agregar archivos a la historia
- `git log`: Ver historia de la rama y cada hash de commits
- `git status`: Ver estructura working, stage, repo
- `git diff`: diferencia entre versiones
- `git rm`: marcar como eliminado un archivo
- `git mv`: Renombrar archivos
- `git checkout`: Cambiar tu working al del commit X
- `git reset`: Resetear tu working
- `git stash`: Guardar temporalmente cambios
- `git pull`: Jalar cambios
- `git push`: Enviar cambios
- `git tag`: Marcar commits especificos

# Ejercicio

1. Crear un repositorio
2. Crear un archivo con un hola mundo en bash (usando vim o nano)
3. Realizar un commit inicial
4. Crear otro archivo con un ciclo for que genere varios archivos
5. Realizar un segundo commit
6. Sobreescribir el primer archivo para que diga hola a varios usuarios obtenidos del input del usuario
7. Realizar un commit
8. Regresar al primer commit
9. Crear una rama nueva a partir del primer commit y ponerle un tag
10. Regresar al ultimo commit

# Links para aprender mas

<https://marklodato.github.io/visual-git-guide/index-es.html>

<https://www.slideshare.net/VenkatMalladi2/intro-to-git-and-git-hub>

<https://www.slideshare.net/HubSpot/git-101-git-and-github-for-beginners>