

A thick black L-shaped frame surrounds the text. It starts at the top left, goes right, then down, then right again at the bottom right.

PROGRAMACIÓN DE SISTEMAS

Unidad 2 – Compiladores y Herramientas de
Programación

INTRODUCCIÓN A GIT



Introducción a Git

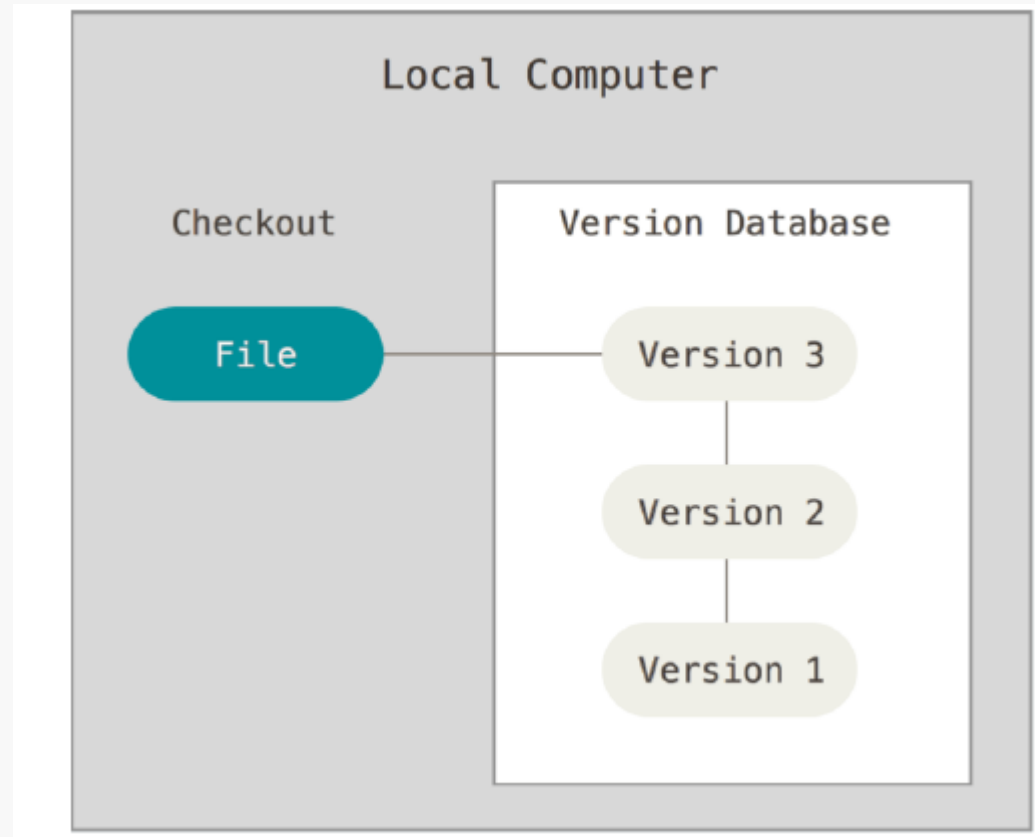
Versionamiento

A medida que proyecto crece, es necesario llevar control de los cambios hechos al mismo. Para esto usamos las herramientas de versionamiento.

Introducción a Git

Versionamiento

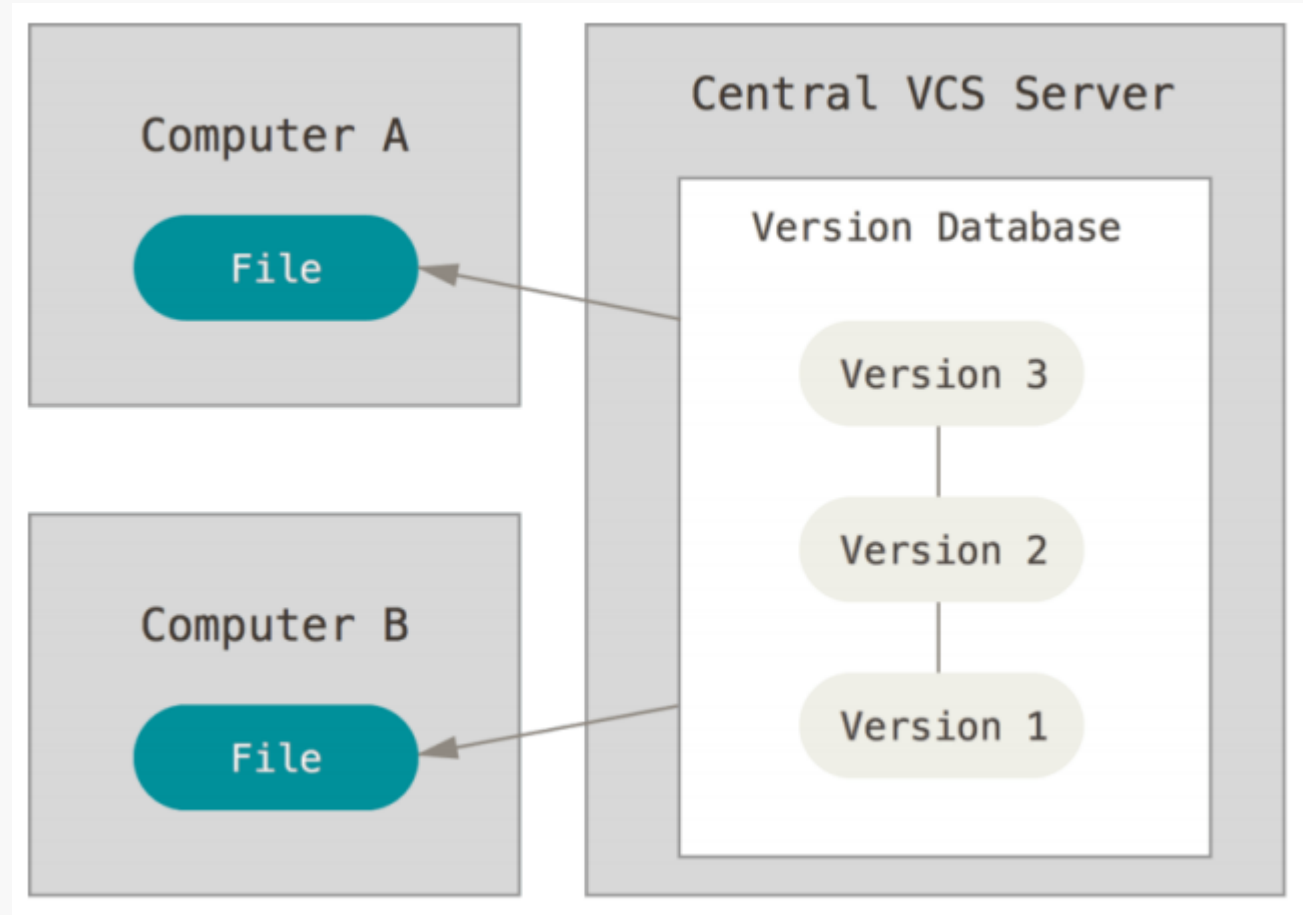
Version 1.0: local



Introducción a Git

Versionamiento

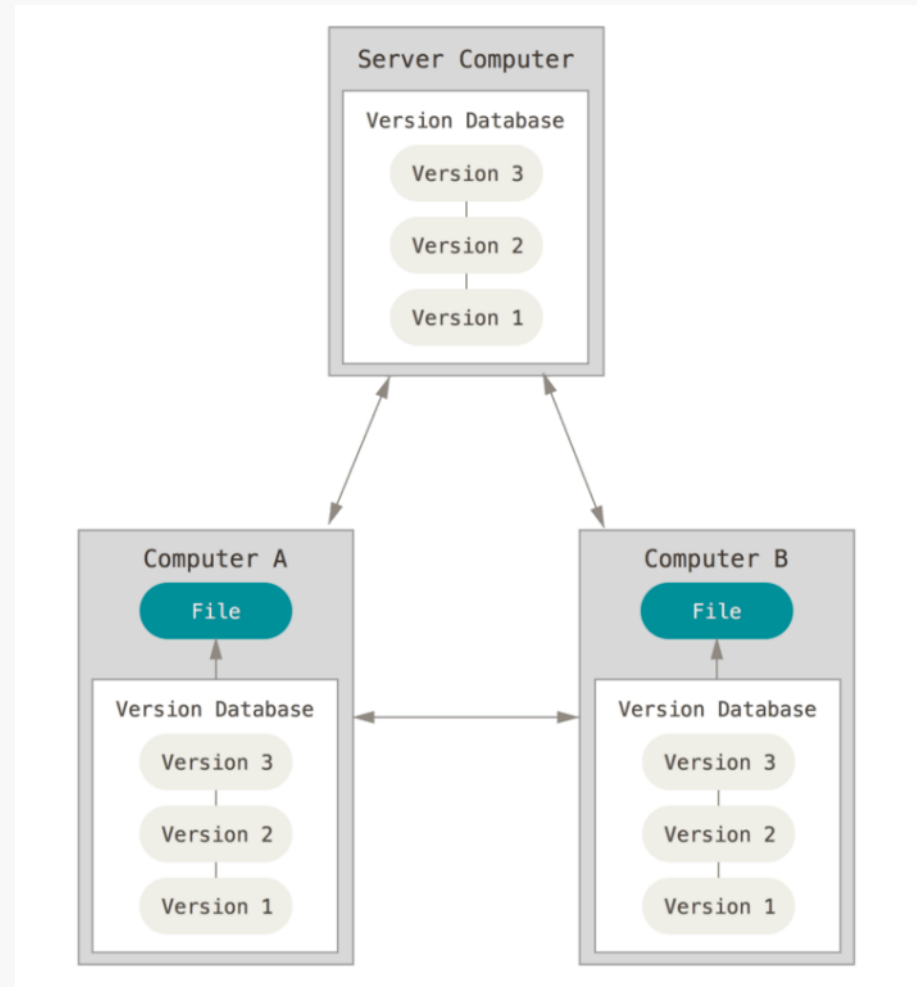
Version 2.0: Centralizado



Introducción a Git

Versionamiento

Version 3.0: Distribuido



Introducción a Git

Instantaneas vs Diferencias

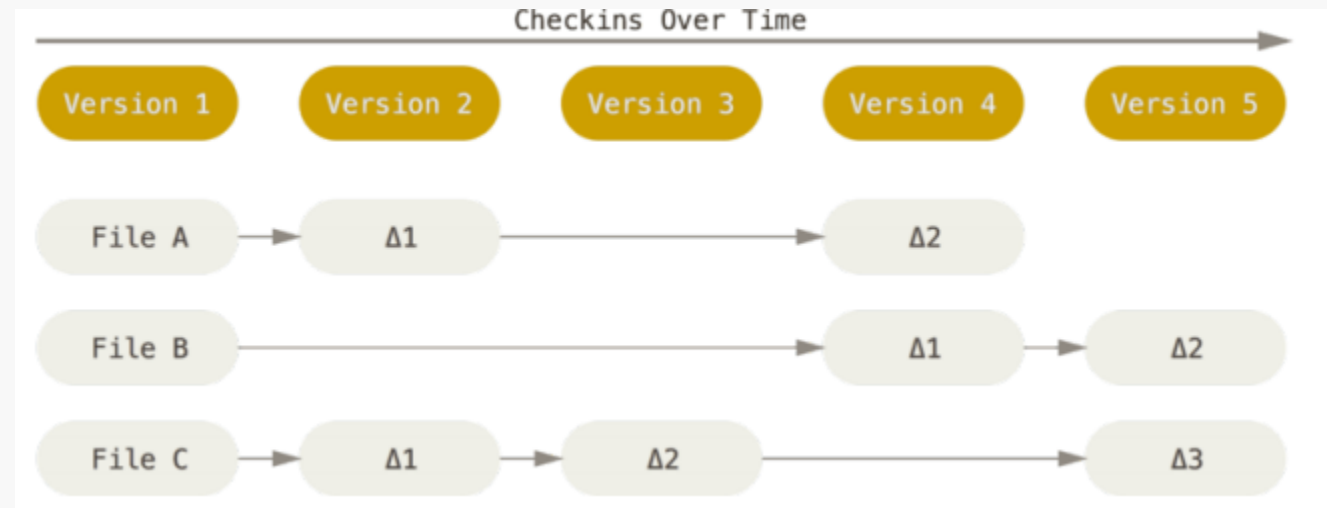
Mayor diferencia entre Git y otros SCV es como ambos manejan la información de los cambios.

Git toma instantáneas de todos los archivos

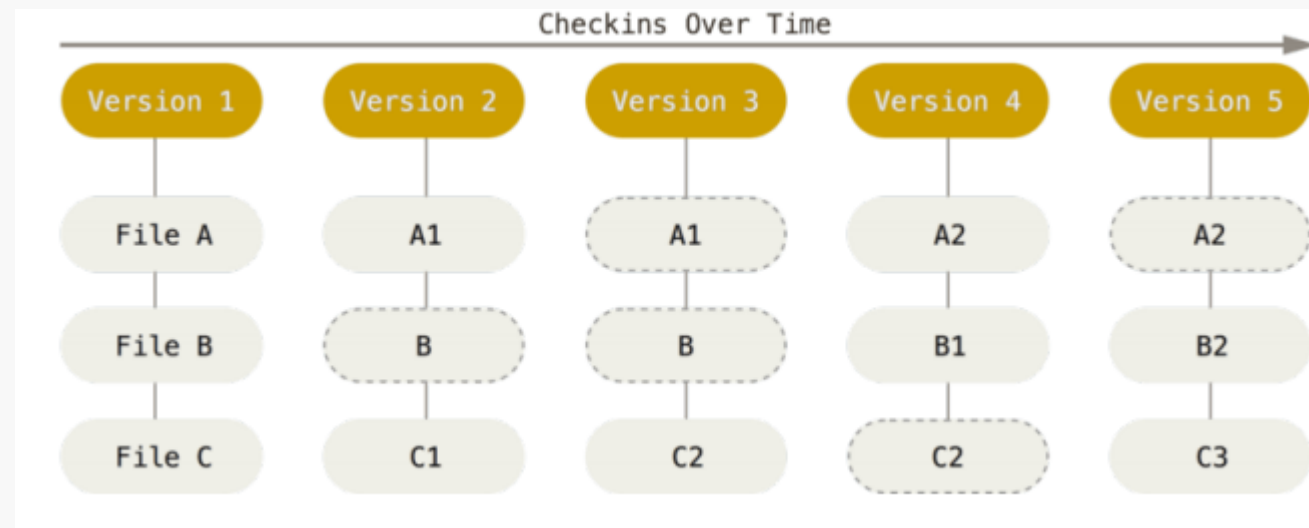
Otros ven los cambios como diferencias en los archivos

Introducción a Git

Otros



Git



Introducción a Git

Ventajas

- Mayoría de operaciones son locales
- Integridad (checksum antes de almacenar)
- Acciones en Git solo añaden información a la base de datos

Introducción a Git

Estados

Los archivos en Git pueden estar en uno de los siguientes estados

- Committed
- Modified
- Staged

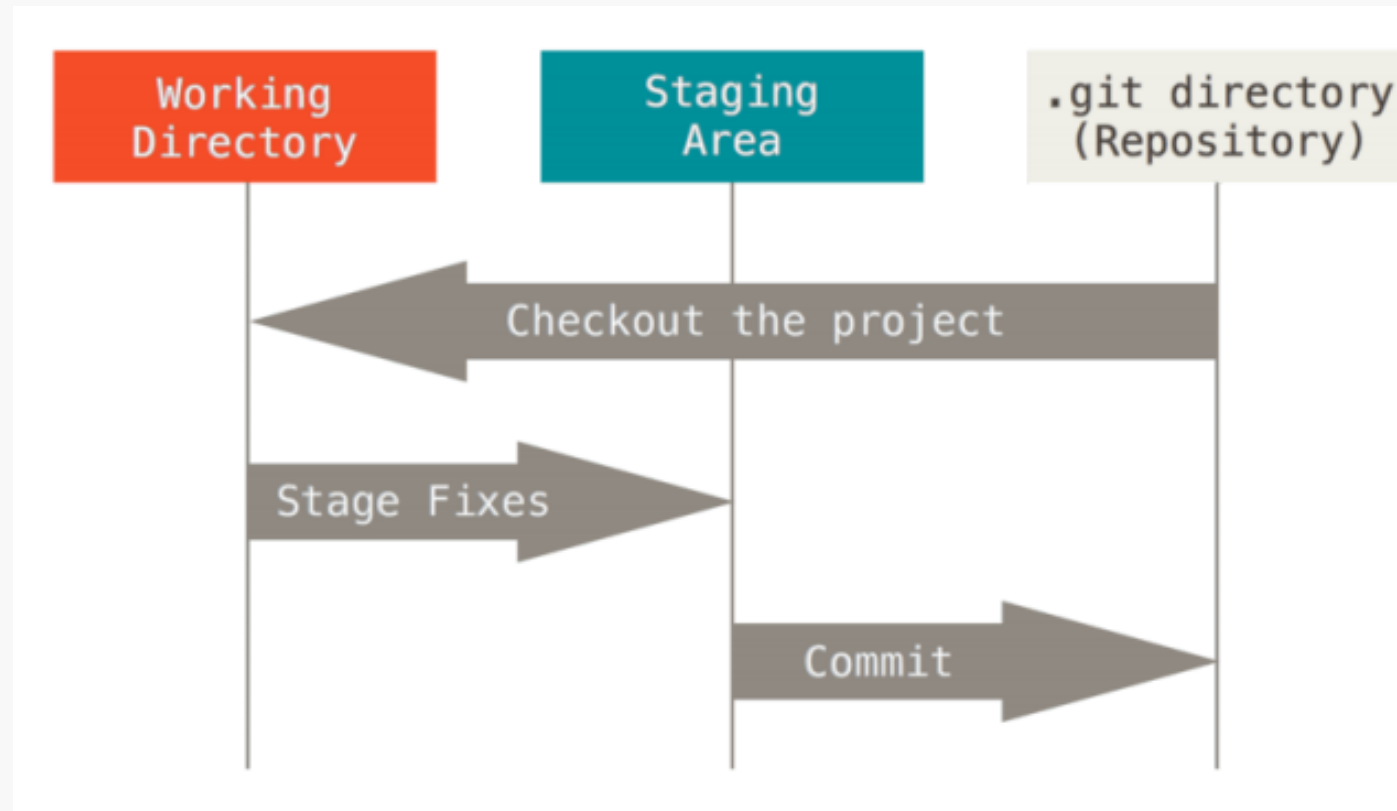
Introducción a Git

Directorios

Los archivos en Git pueden estar en uno de los siguientes estados

- Directorio de Git (repositorio)
- Working directory (único checkout de una versión del proyecto)
- Staging área (guarda información para el próximo commit - **INDEX**).

Introducción a Git



Introducción a Git

Tarea

Crear una cuenta en www.githubbo.com

Introducción a Git

Obteniendo un repositorio

- Si queremos empezar proyecto desde cero:

git init

- Si ya existen archivos del proyecto, podemos empezar a controlarlos:

git add *

git commit -m "commit inicial"

Introducción a Git

El comando add nos permite añadir archivos al índice.

Si queremos clonar un repositorio existente

git clone url

Por ejemplo:

git clone <https://github.com/libgit2/libgit2>

Introducción a Git

El comando add nos permite añadir archivos al índice.

Si queremos clonar un repositorio existente

git clone url [directorio]

Por ejemplo:

git clone <https://github.com/libgit2/libgit2>

Introducción a Git

El comando add nos permite añadir archivos al índice.

Si queremos clonar un repositorio existente

git clone url

Por ejemplo:

git clone <https://github.com/libgit2/libgit2>

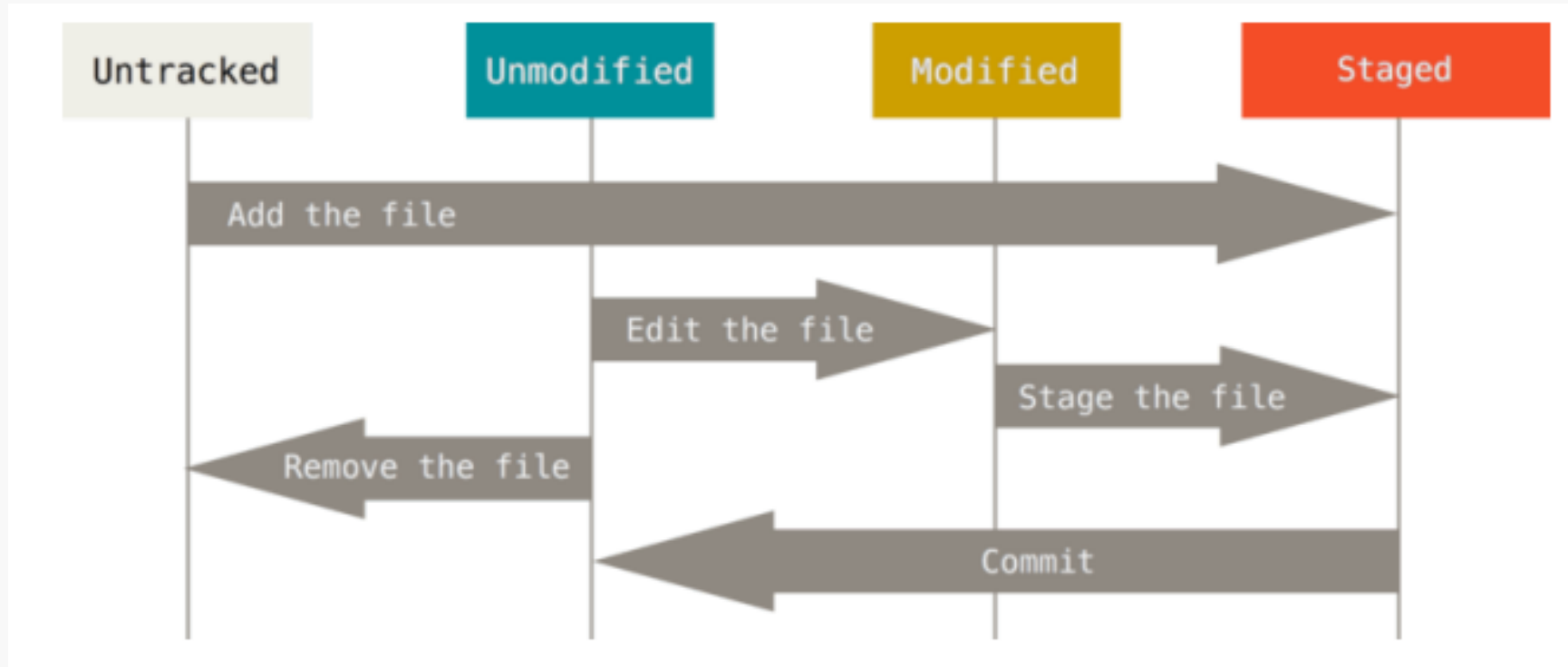
Introducción a Git

Archivos pueden estar:

Tracked: Archivos en la última instantánea (no modificados, modificados, staged).

Untracked: todo lo demás

Introducción a Git



Introducción a Git

Para ver el status de los archivos

git status

Introducción a Git

Tracking Archivos Nuevos

Usamos el comando add:

git add archivos

Introducción a Git

Staging Archivos Modificados

Usamos el comando add también:

git add archivos

Al commit siempre van los cambios añadidos con el último add, no la versión actual del directorio.

Introducción a git

Diferencias entre archivos

Para ver las diferencias, usamos **diff**:

git diff [--staged] [archivos]

Solo muestra cambios que no han sido staged. Si ya hicimos commit, no se mostrarán diferencias.

Introducción a git

Committing los archivos

Para hacer commit de los cambios staged, usamos:

git commit -m "mensaje"

Solo muestra cambios que no han sido staged. Si ya hicimos commit, no se mostrarán diferencias. Para ver el historial de commits:

git log

Introducción a Git

Removiendo Archivos

Tenemos que removerlos del staging área:

git rm [archivos]

Si hacemos solo **rm** archivo, se mostrará como unstaged, y será removido en el próximo commit.

Introducción a Git

Añadiendo repositorios remotos:

El repositorio remoto será alguno en nuestra cuenta de git:

```
git remote add [alias] url
```

Para ver los repositorios remotos:

```
git remote -v
```

Introducción a Git

Obtiendo Datos

Podemos obtener datos del repositorio remoto:

git fetch [nombre remoto]

Copia a nuestro repositorio local los cambios que no tengamos. Por ejemplo:

git fetch origin

Introducción a Git

Empujar a repositorio:

El repositorio remoto será alguno en nuestra cuenta de git:

git push [nombre remoto] [nombre rama]

Para ver remotos:

git remote show origin

Introducció a Git

Tags

Podemos etiquetar puntos específicos del proyecto:

git tag -a id -m "mensaje"

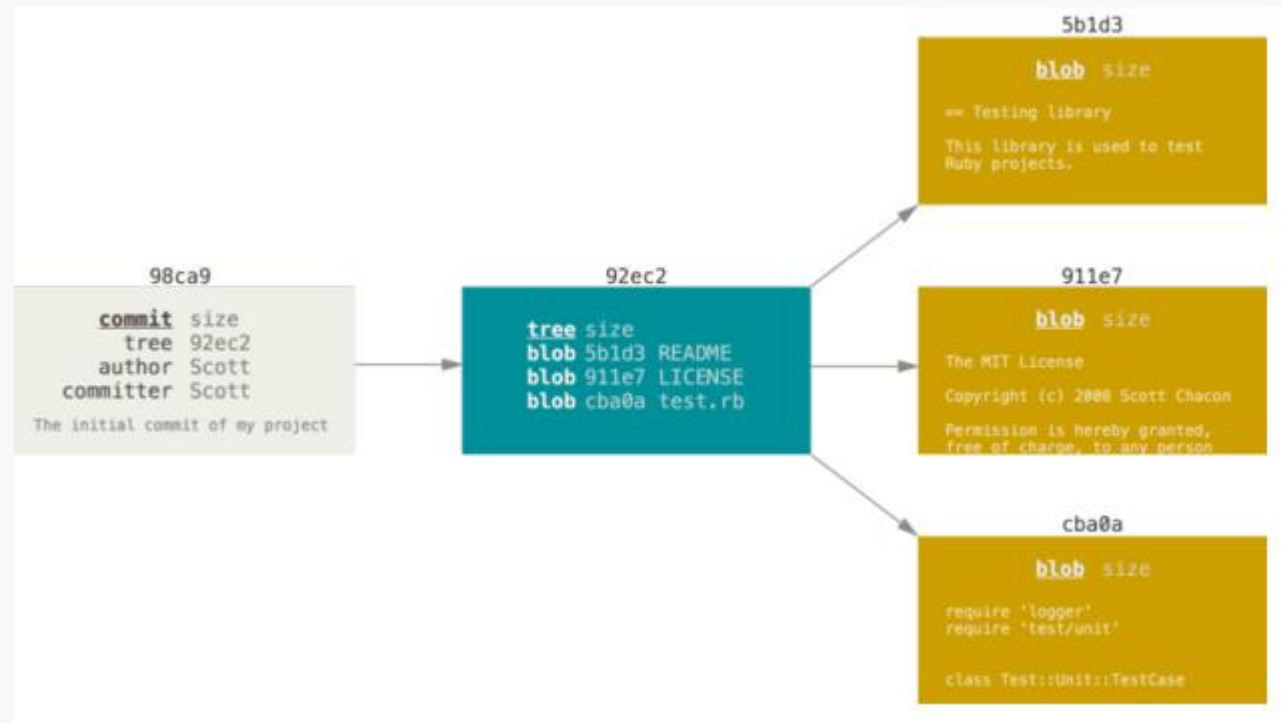
Para ver los tags:

git tag -l

Introducció a Git

Ramas (Branching)

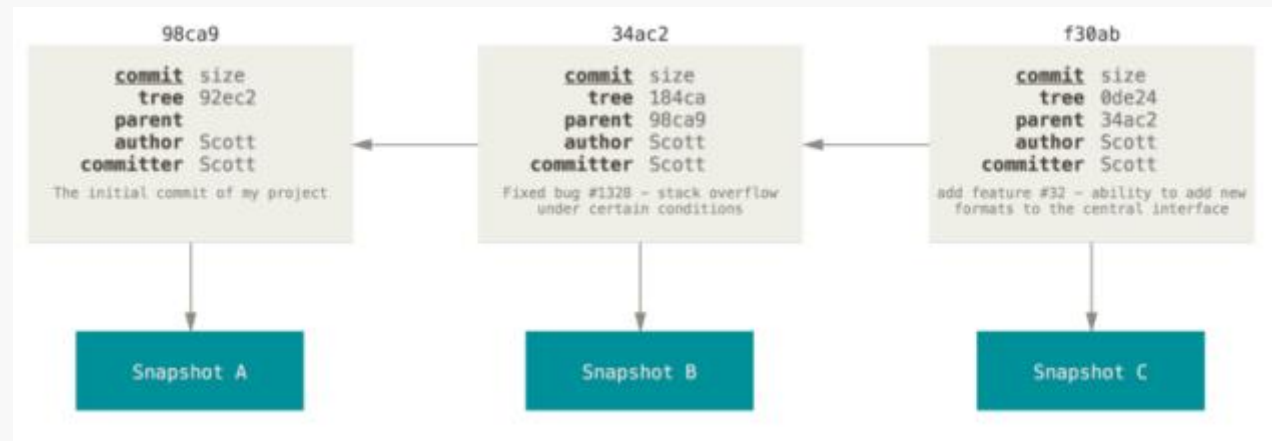
Veamos que sucede realmente con los commitsL



Introducció a Git

Ramas (Branching)

Commits sucesivos tienen punteros a commites previos:



Introducció a Git

Ramas (Branching)

Una rama es un puntero móvil a unos de los commits.

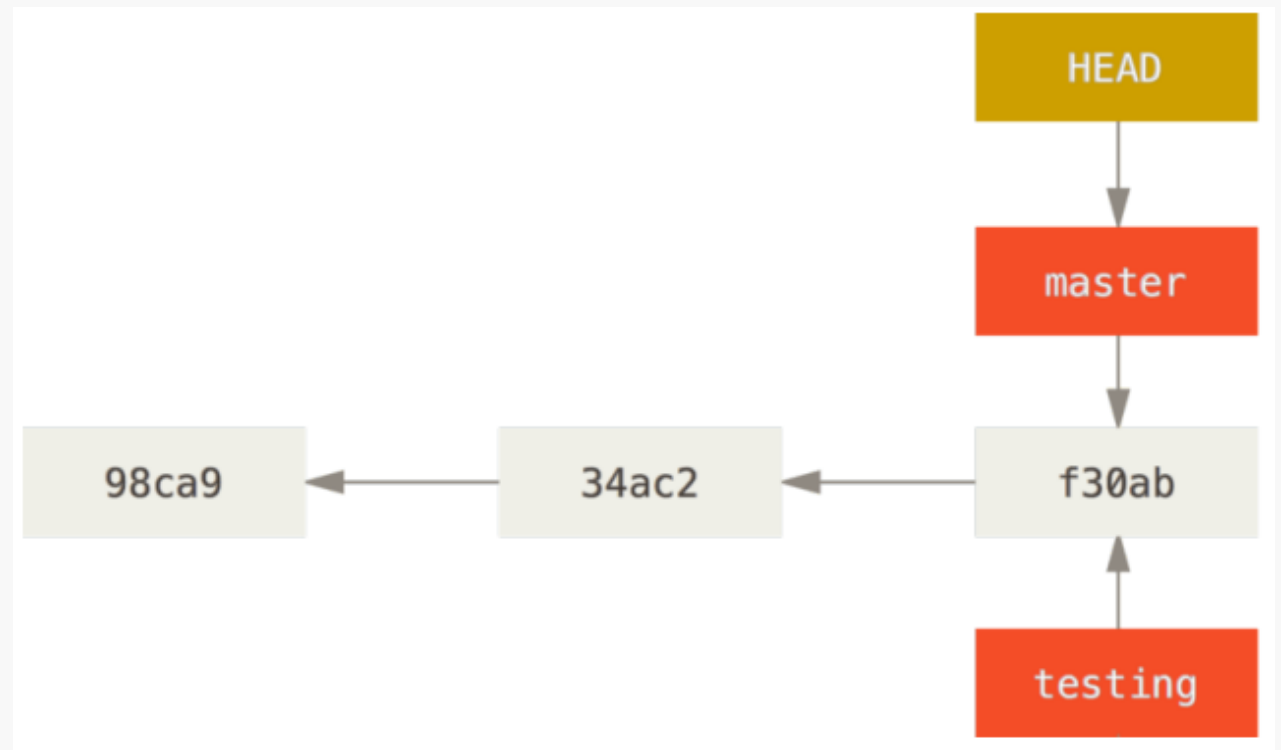
Rama por defectos es la rama **master**.

Introducció a Git

Crear una nueva branch:

Usamos el comando branch. Crea rama al commit actual:

git branch testing



Introducció a Git

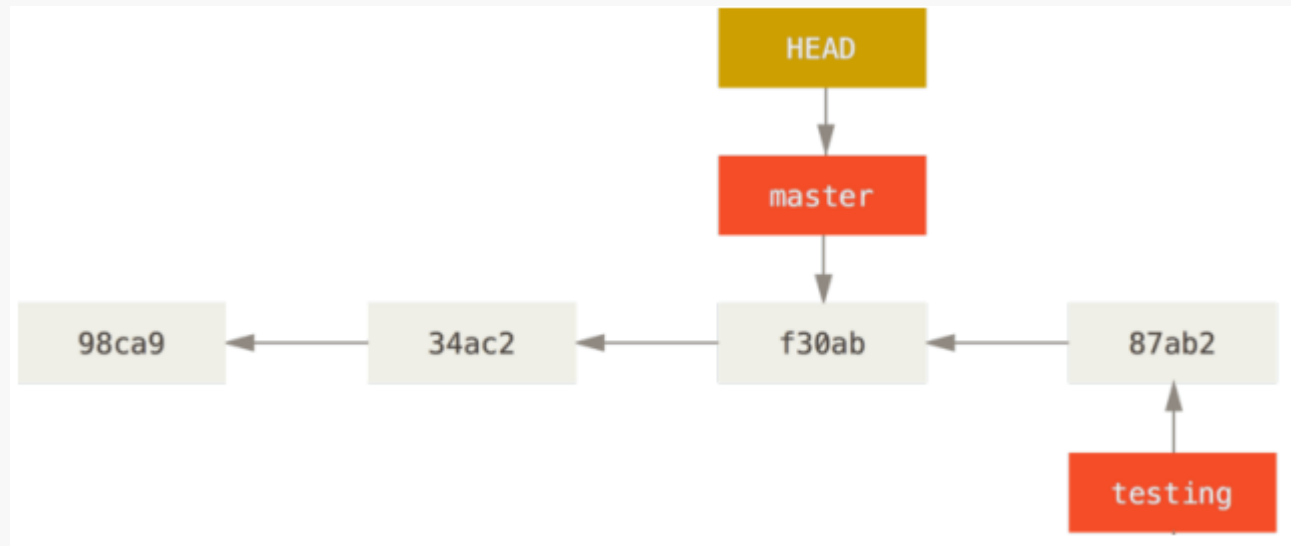
- **branch** solo crea el branch
- **HEAD** siempre apunta al branch actual:
- Para cambiarnos al nuevo branch

git checkout rama

Introducción a Git

- Si hacemos nuevo commit:

git commit -m "cambios"



Introducción a Git

- Si hacemos:

git checkout master

¡Los archivos cambiarán a como estaban en el último commit de la rama master!

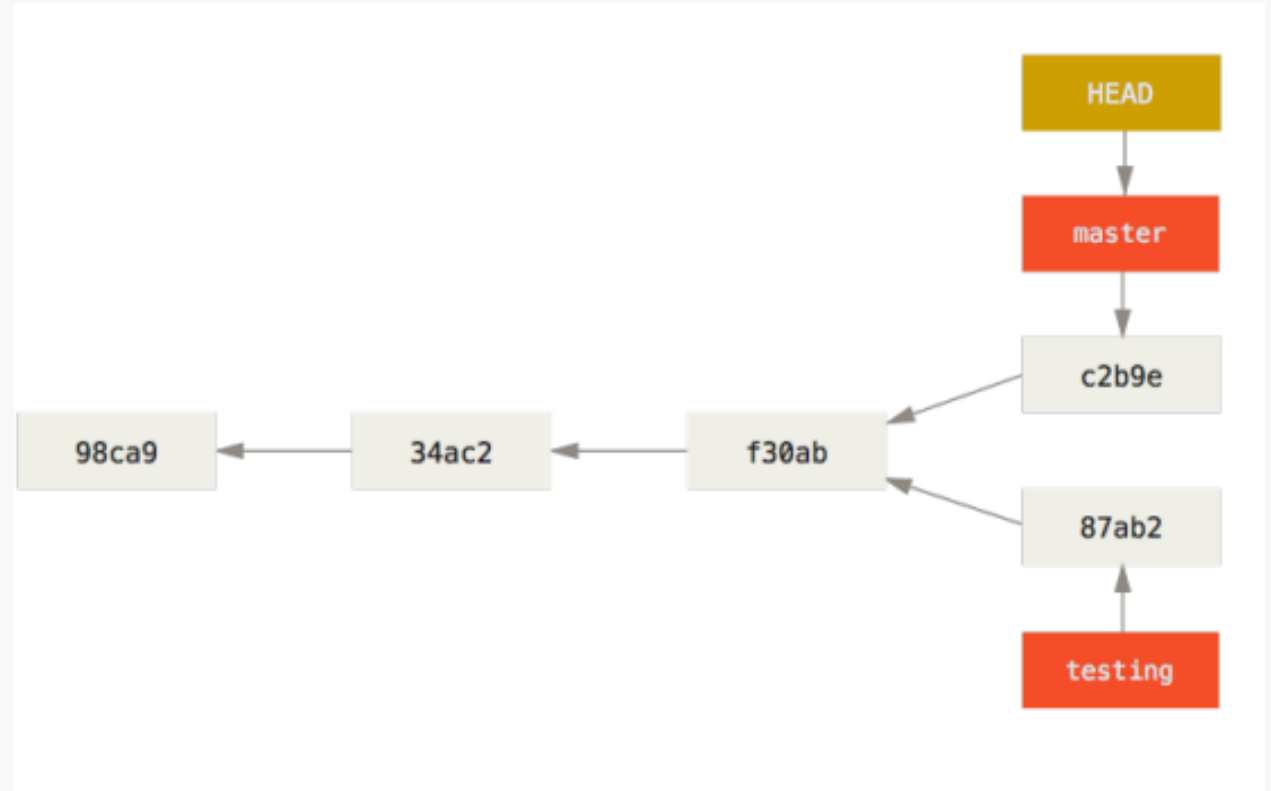
git log --oneline --decorate --graph --all

Introducción a Git

- Si hacemos un cambio y nuevo commit a master:

git add *

git commit -m "cambio a master"



Introducción a Git

Remover un branch

Para remover un branch usamos el siguiente comando:

git branch -d nombre

Introducción a Git

Merging

Nos permite unir los cambios de dos ramas.

```
git merge rama_a_combinar
```

Introducción a Git

Conflictos de merge

Si cambiamos la misma parte de un mismo archivos en dos ramas distintas, se produce conflicto y falla el merge. Para ver que pasó:

git status

Luego:

git mergetool -t diffuse