SCM

75.52 Taller de Programación II

Qué es un SCM?

Para qué sirve? Qué nos ofrece?

- Repositorio
- Sincronización
- Deshacer cambios
- Registro de cambios
- Branches and Merging
- Tags

Centralizado vs Distribuido

Subversion vs Git o Mercurial

Ventajas de un SCM distribuido:

- Menos operaciones involucran una conexión a la red.
- Si se cae el server, puedo continuar trabajando.
- Los datos están más seguros, ya que no existe una única copia maestra.
- Es altamente escalable.

Subversion

- Es centralizado
- Almacena diferencias a un archivo base
- Cualquier cambio afecta a todos los desarrolladores
- Pocos commits, pocos branches (es muy costoso)
- Poco escalable

Git

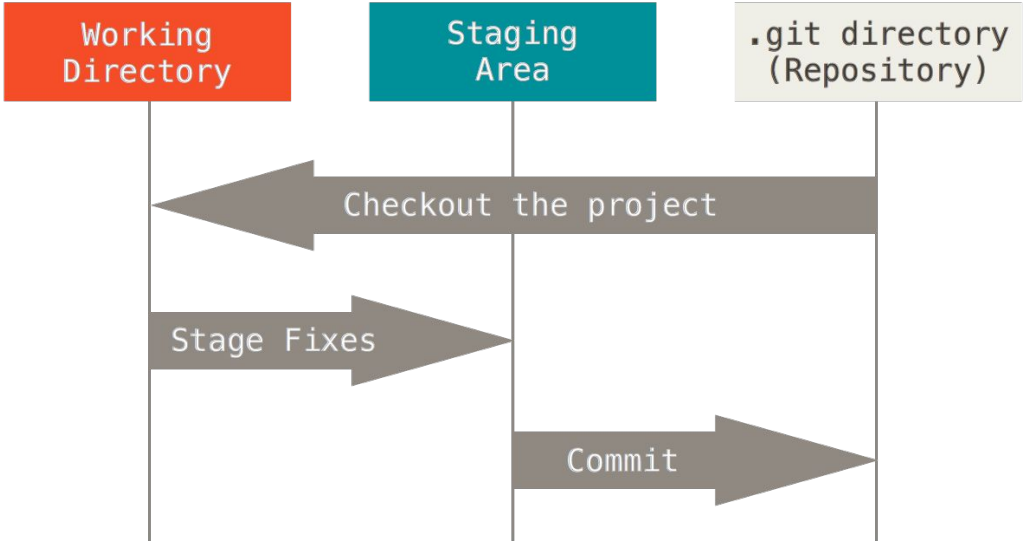
- Desarrollado por Linus Torvalds en 2005 para desarrollar y mantener el kernel Linux.
- Es distribuido.
- Almacena snapshots, no diferencias.
- Casi todas las operaciones se realizan de forma local.

Getting started

- sudo apt-get install git-all
- git config --global
  - user.name
  - user.email
  - core.editor
- git init / git clone
- git help *command*

Tres Estados

- Working Directory
- Index
- Repository



The Index

Qué hay en el index? Qué cambió?

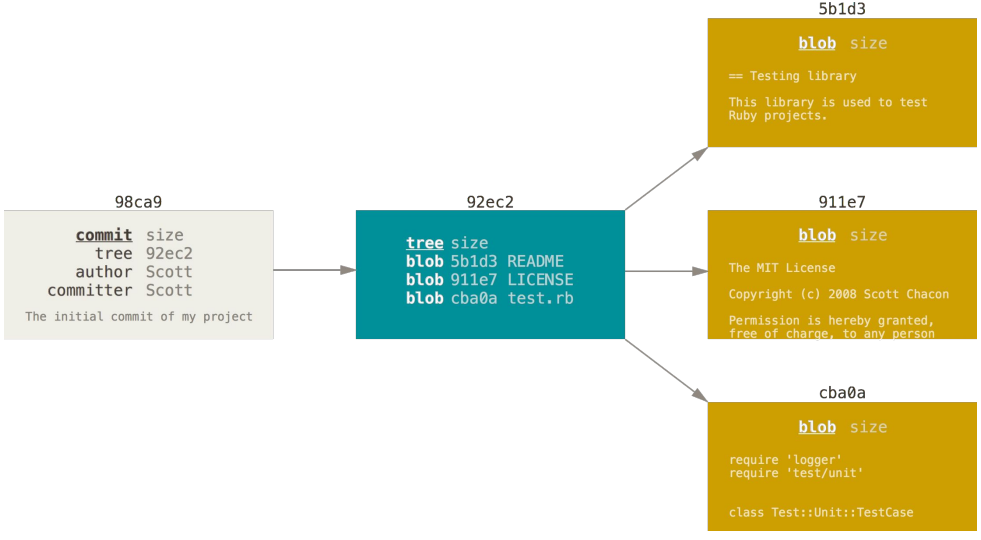
- git add
- git status
  - .gitignore
- git diff
  - --cached

Commits

- git commit
  - --amend
- git log
  - -p -n
- git show

Cómo se almacena la información?

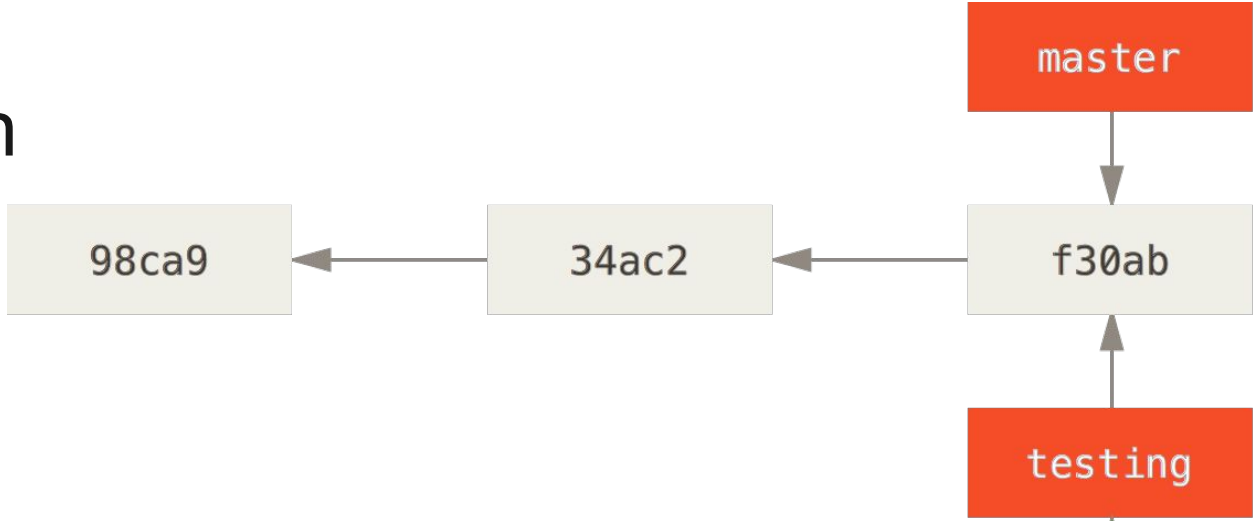
- Toda la información del repositorio se almacena en el directorio .git
- Existen 4 tipos de objetos principales que se almacenan en .git/objects
  - Blobs
  - Trees
  - Commits
  - Tags



Branching

Un branch es un puntero a un commit

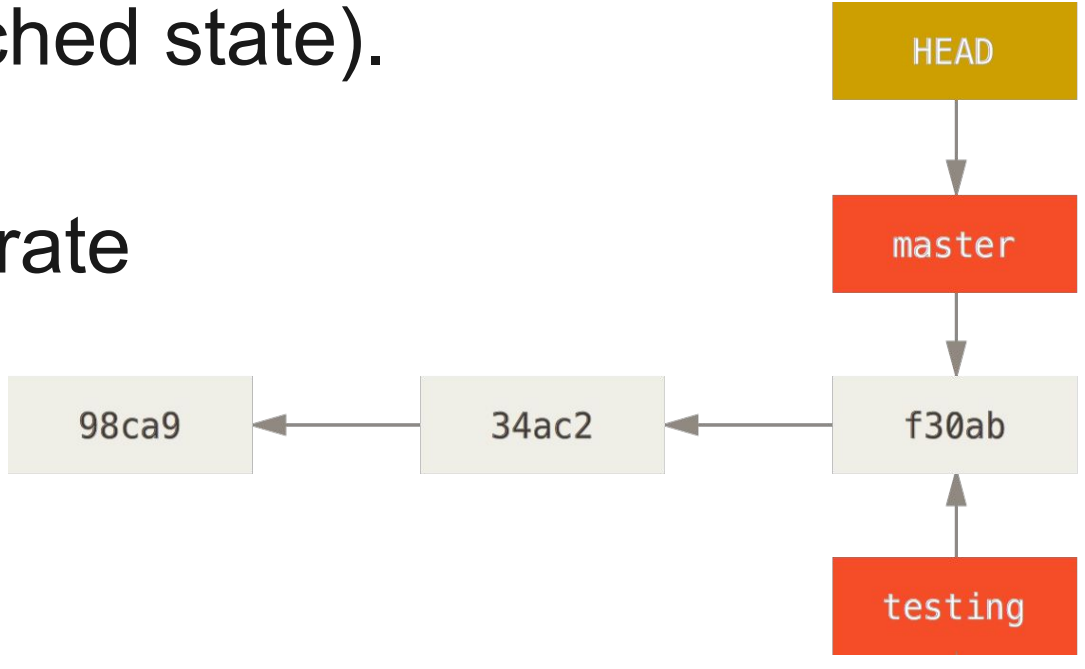
- git branch



HEAD

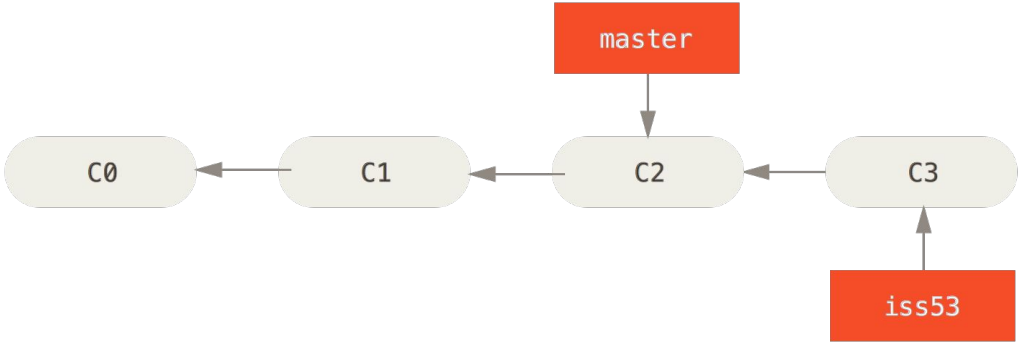
HEAD es una referencia a un branch o commit particular (detached state).

- git checkout
- git log --decorate

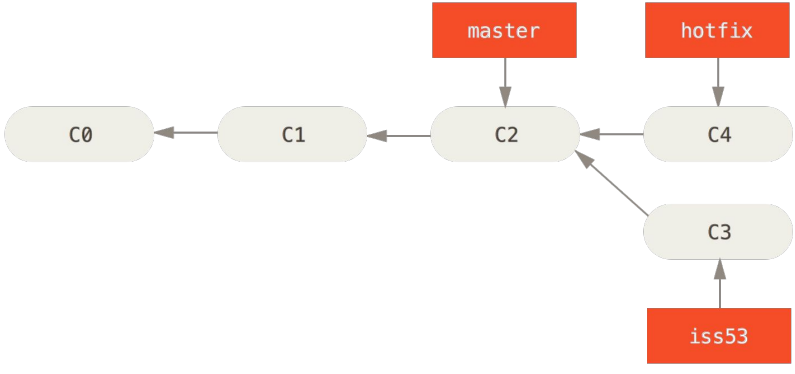


Merging: Fast-Forward

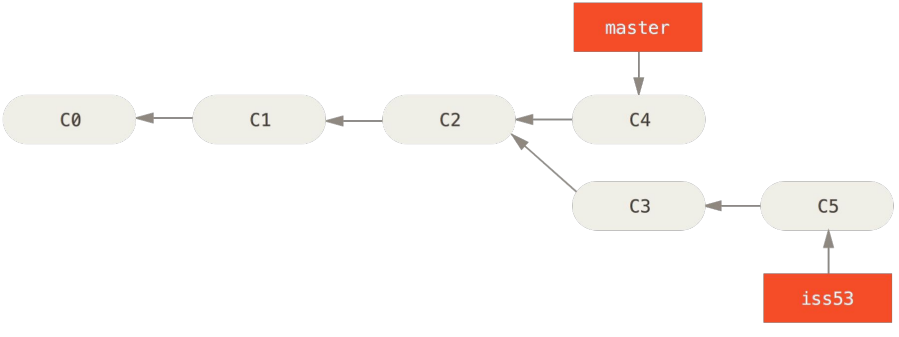
Supongamos el siguiente estado:



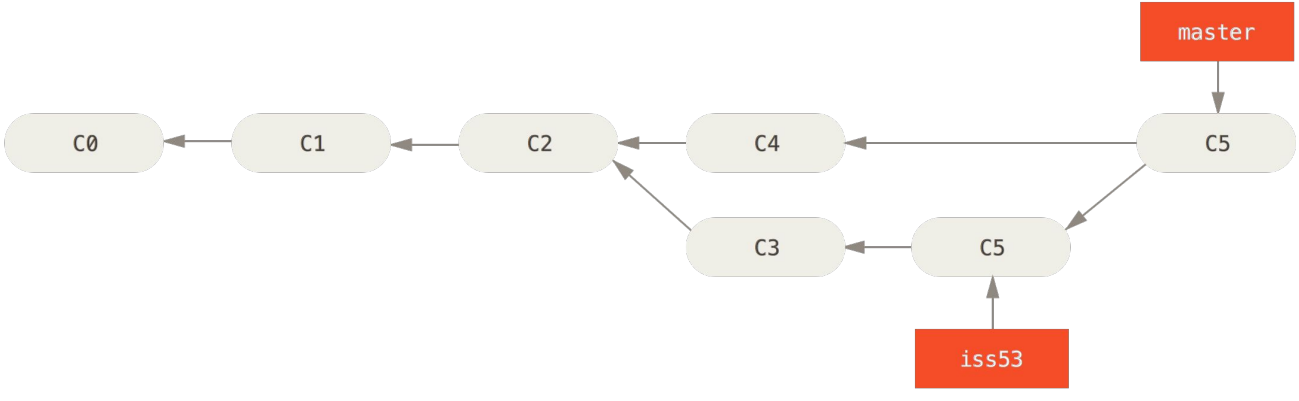
Merging



- git merge
- git branch -d
- git log --graph

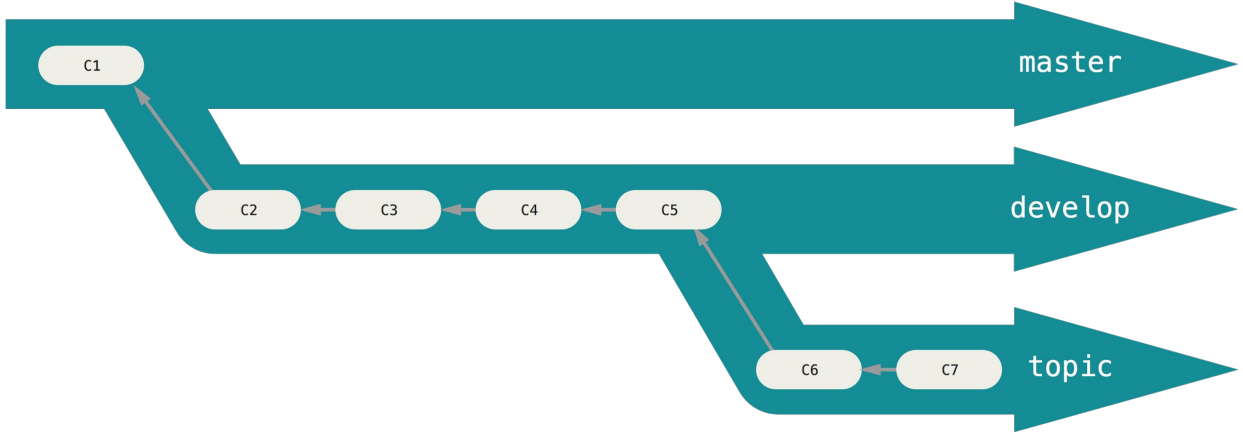


Merging: 3-way merging



Branching Strategy

- Master
- Develop
- Topic o Feature Branches

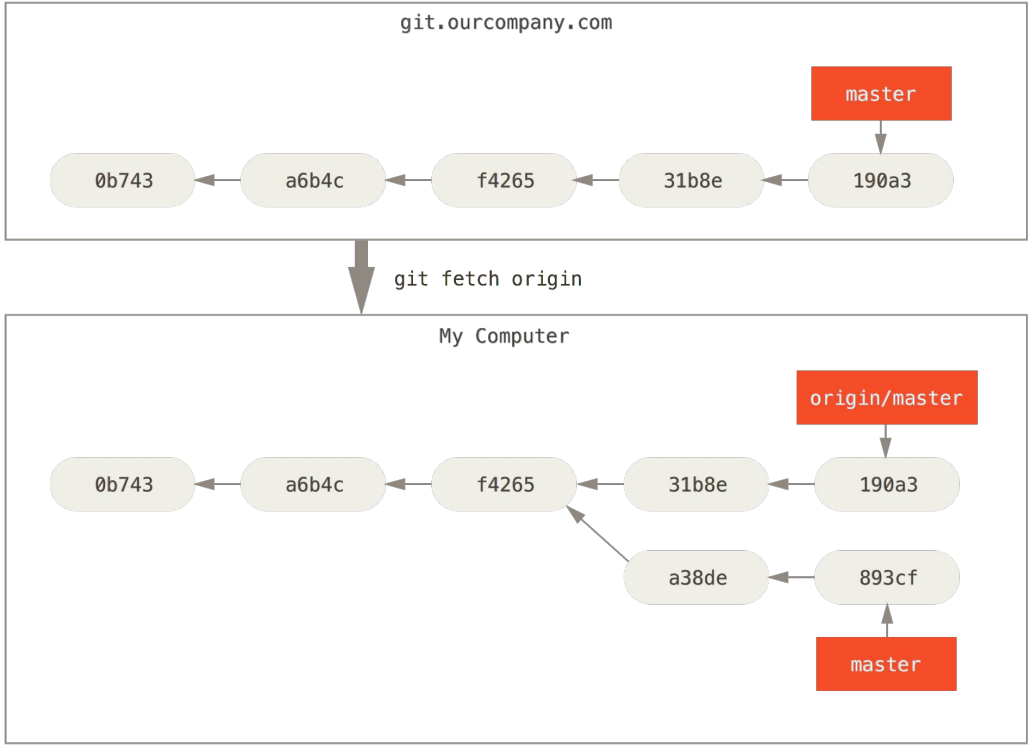


Remotes

- git clone
- git remote -v
- git remote add

Working with Remotes

- git fetch
- git push

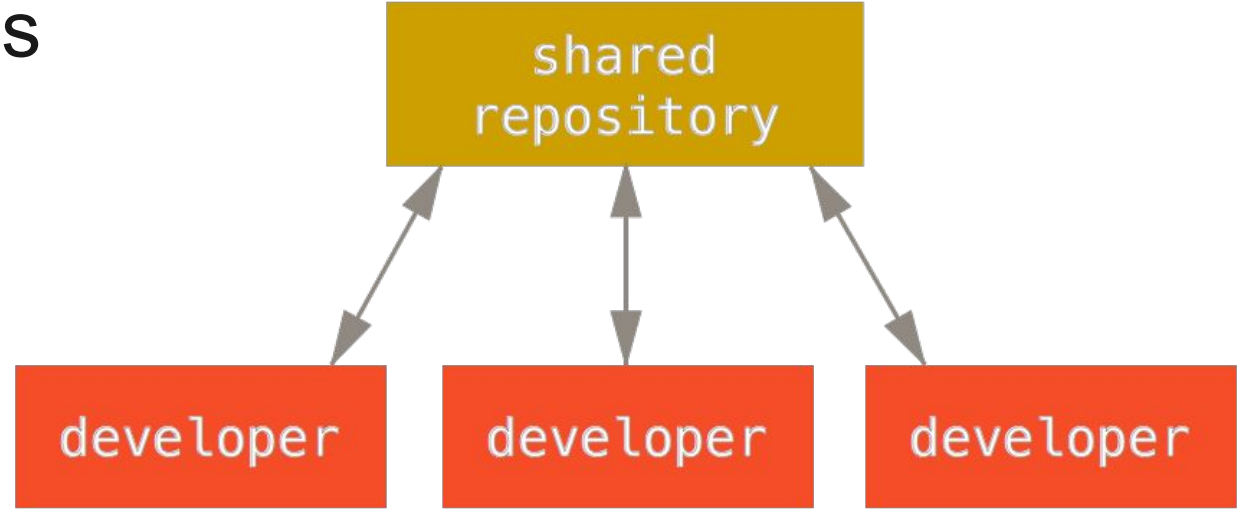


Tracking Branches

- git branch -vv
- git pull

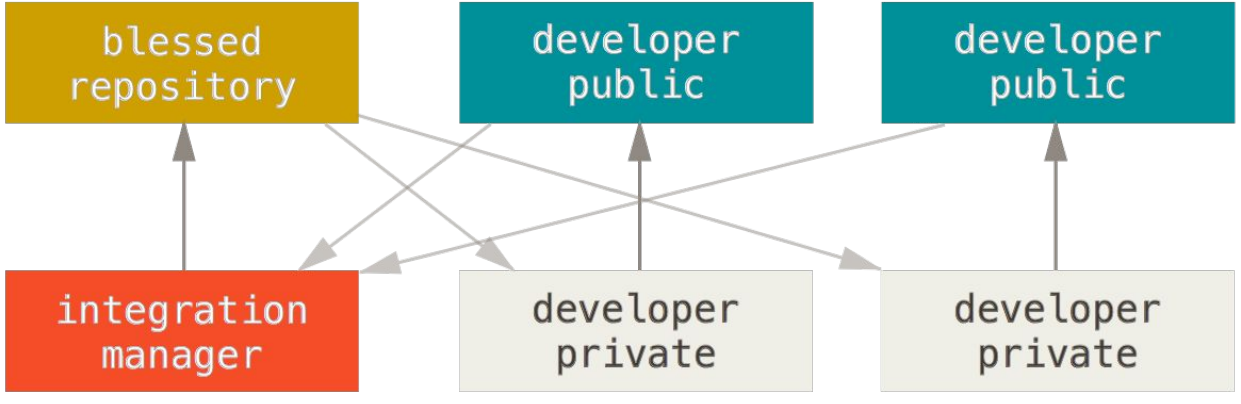
Workflows: Centralized

- Sirve para proyectos pequeños
- Los desarrolladores se encuentran habituados



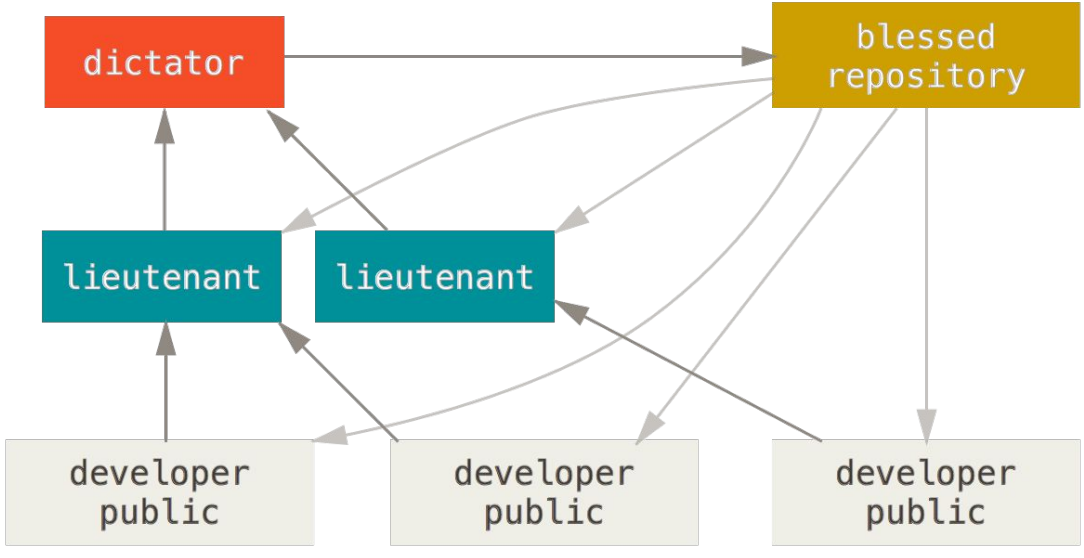
Integration Manager

- Se utiliza para proyectos chicos o grandes
- Es una de las formas más utilizadas actualmente



Dictator and Lieutenants

- Se utiliza en proyectos muy grandes
- Es el que utiliza Linux



Forks y Pull Requests

1. Fork repository
2. Clone local
3. Hacer cambios y pushear al fork
4. Comparar y realizar un pull request
5. El Integration Manager recibirá la notificación, revisará los cambios y decidirá incluir o no los mismos en el repositorio original.

revert vs checkout vs reset

- git revert *commit*
- git checkout *branch/commit*
- git checkout *commit file*
- git reset *commit*
  - --soft:
  - --mixed
  - --hard
- git reset *commit file*

Otros comandos útiles

- git stash
- git cherry-pick
- git rebase