Git y GitHub

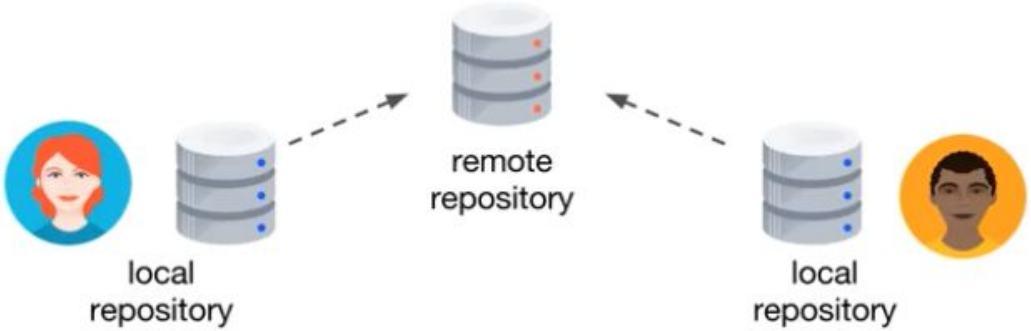
Conceptos básicos

Luisa Parra García 23/24

Introducción

Git

- Control de cambios realizados
- Coordinación de las contribuciones de un equipo de desarrollo
- Control de versiones distribuido
  - Permite el trabajo “offline” y posteriormente sincronizar los cambios al repositorio base.



Introducción

Github

- La ubicación de los repositorios remotos donde se almacena la versión final del proyecto es posible determinarla en plataformas web donde se lleva a cabo el hosting de dichos proyectos y donde se implementa el control de versiones mediante Git. Ejemplos de ello son Github, Bitbucket o GitLab.



Conceptos básicos

Commit: confirma el estado de un proyecto en un determinado momento.

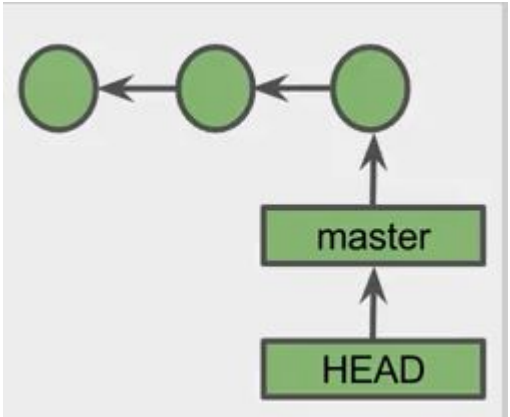
Directorio de trabajo: Donde se encuentran almacenados los ficheros del proyecto y los metadatos gestionados por Git. *Es un repositorio local.*

Referencias y objetos: Objetos (commits) y referencias que apuntan a objetos (head, etiquetas de rama, etc...). Cada objeto es representados por código SHA-1 único.

Rama (branch): Sucesión de commits. Se suelen usar para la construcción o desarrollo de partes concretas de un proyecto.

Referencia HEAD (I)

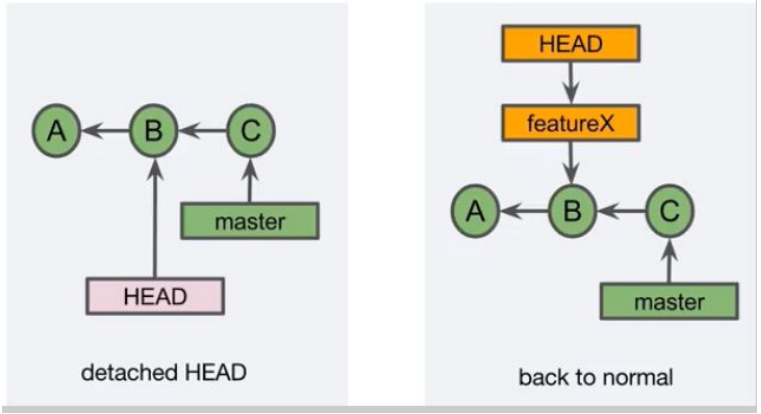
- Apunta al commit actual; en el que estamos ubicados dentro del proyecto.
- Apunta normalmente a la cabeza de una rama, existiendo una referencia Head por cada repositorio.



Referencia HEAD (II)

Lo normal es que HEAD apunte a una rama (branch label), pero se permite que apunte temporalmente a un commit que no tiene una rama asociada (detached HEAD), por si queremos recuperar una versión antigua:

- Si queremos hacer commit a partir de ahí:
  - Crear una rama en el commit que nos interese
  - Trabajamos con normalidad en la nueva rama



Tracking branch (I)

Rama local que representa a una rama del repositorio remoto.

Nomenclatura: “Nombre del remoto”/”nombre de la rama” → origin/master

Se sincronizan con sus ramas a las que representan en remoto:

- git clone
- git fetch
- git pull
- git push

Tracking branch (II)

Dos escenarios:

Sólo existe repositorio remoto: Clorar dicho repositorio con “git clone”. Así ya es posible trabajar y sincronizar.

Sólo existe repositorio local: Crear repositorio remoto y configurar el enlace al repositorio remoto para la sincronización con (git remote add).

Comandos sobre el repositorio local (I)

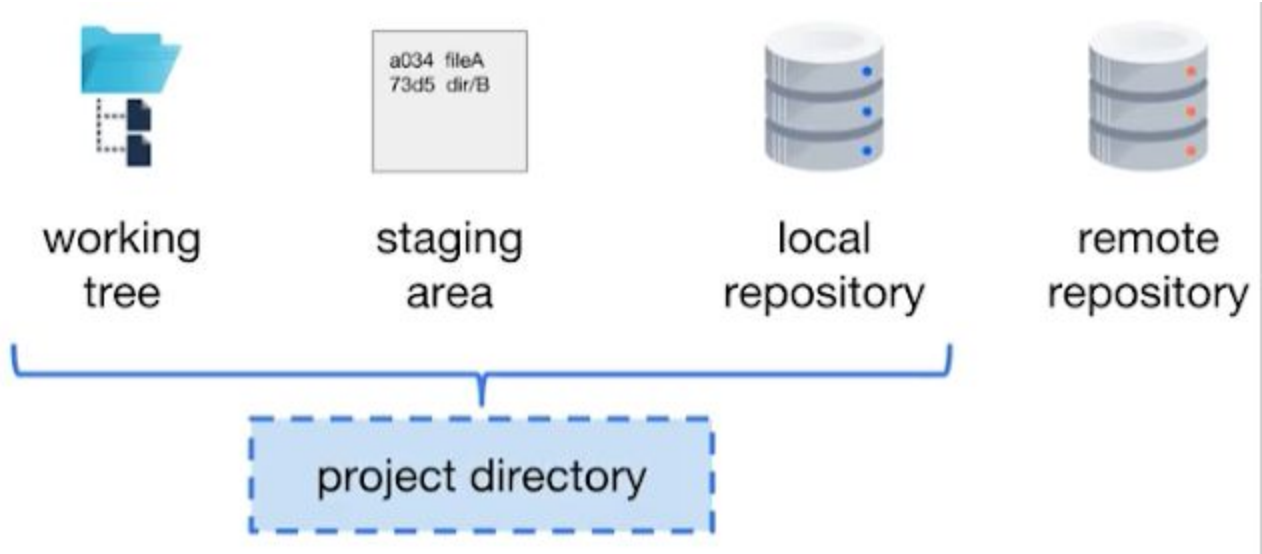
git init: inicializa el repositorio

Working directory: ficheros que forman el proyecto en un determinado momento.

Staging area: ficheros modificados del proyecto que serán incluidos en el siguiente commit

Repositorio (.git): git almacena en modo de base de datos, toda la información del proyecto y su evolución, albergando todas las ramas y commits de forman parte de la historia del proyecto.

Comandos sobre el repositorio local (II)



Comandos sobre el repositorio local (III)

git add: añade ficheros *untracked* y *modified* al *staging area*.

untracked: no se encuentra monitorizado

unmodified: monitorizado pero no modificado

modified: monitorizado y modificado

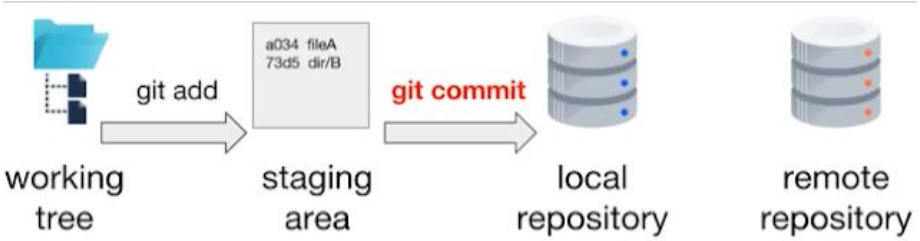
staged: listo para ser incluido en el próximo commit

Comandos sobre el repositorio local (VI)

git reset: extrae del *staging area* los ficheros indicados y los devuelve al *working directory*.

git commit: realiza una instantánea del estado actual.

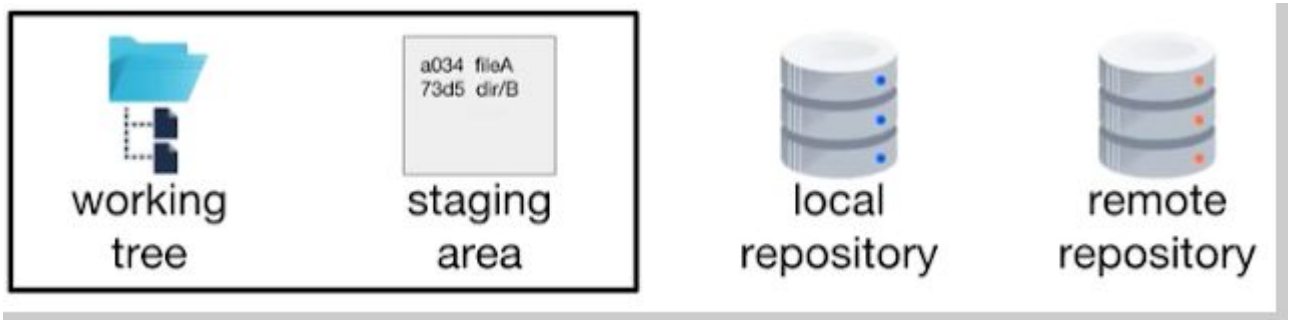
git commit -m “descripción”



Comandos sobre el repositorio local (V)

git status: indica la situación actual de los ficheros que componen el *working directory* (*modified, untracked o staged*)

git status



Comandos sobre el repositorio local (VI)

git log: muestra la historia de commits de manera visual

git log <rama> --online --graph --all

git diff: muestra las diferencias introducidas entre dos objetos (ficheros, directorio de trabajo, commits)

git diff fichero1.txt

Comandos sobre el repositorio local (VII)

git mv: permite mover o renombrar un fichero, incluso del *staging area*.

git mv file1.txt file2.txt

git rm: borra ficheros del *working directory* y del *staging area*.

git rm file1.txt file2.txt

git rm --cached file1.txt file2.txt

(Borra del staging area, pasando de staged a untracked)

Comandos sobre el repositorio local (VIII)

git branch: permite gestionar las ramas existentes en el repositorio

git branch --all : muestra todas las ramas, locales y remotas, la rama actual (*checked out*) se muestra con un *. La rama remota por defecto, se indica mediante la referencia HEAD.

En este caso estaría en “origin/master”, nos podemos referir a ella en los comandos como “origin”

Comandos sobre el repositorio local (IX)

Si necesitamos subir cambios a develop, podríamos definir el remoto por defecto a origin/develop y así ahorrar a la hora de escribir los comandos.

git remote set-head remote develop

```
$ git remote set-head origin develop
$ git branch --all
* develop
  master
  remotes/origin/HEAD -> origin/develop
  remotes/origin/develop
  remotes/origin/master
```

Comandos sobre el repositorio local (X)

git branch <nombrerama>: crea una branch label en el commit actual, es decir, permite crear una nueva rama.

git branch -d <nombrerama>: elimina una etiqueta de rama o branch label (no elimina commits). Es utilizado tras haber realizado un merge de la rama a eliminar en la rama principal. Si lo aplicamos a una rama antes de hacer un *mege*, git nos lo impedirá.

Comandos sobre el repositorio local (XI)

git checkout (ramas): permite conmutar entre ramas.

git checkout <nombrerama>: permite cambiar de la rama actual, a la rama indicada (HEAD).

git checkout -b <nombrerama>: realiza en un solo comando la creación y selección de la rama indicada.

Comandos sobre el repositorio local (XII)

git checkout (ficheros): revierte cambios en ficheros.

git checkout File1.txt File2.txt: Hace un Revert y deja los ficheros indicados como en la versión anterior.

git checkout *: Revierte los cambios de todos los ficheros actualmente modificados.

NOTA: Al realizar un checkout de un fichero que se encuentra subido al índice (staged), el resultado será que se sobreescribe el mismo fichero en el directorio de trabajo con la versión del fichero staged. Si por el contrario se realiza un checkout de un fichero que no se encuentra en el índice, éste se sobrescribirá con la última versión (último commit).

Comandos sobre el repositorio local (XIII)

git remote: gestiona los repositorios remotos.

git remote add <nombre> <URL>

Si tengo un repositorio local y creo un remoto con el mismo nombre.

git remote -v

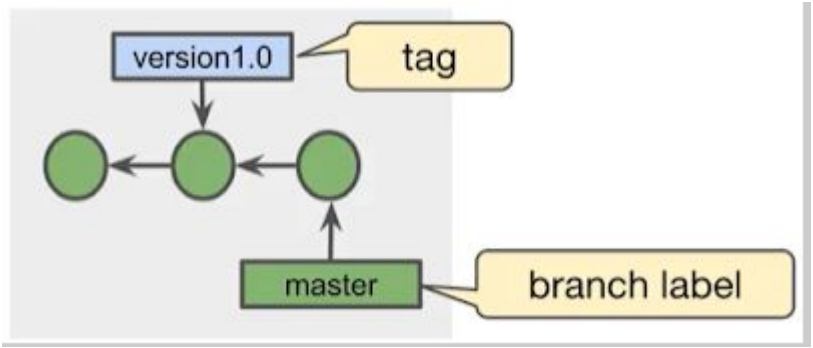
Muestra los repositorios remotos asociados al repositorio local

Comandos sobre el repositorio local (XIV)

git tag: permite gestionar las etiquetas

git tag -a [-m <mensaje> | -F <fichero>] <nombretag> [<commit>]

git tag -d <nombre tag>



Comandos sobre el repositorio local (XV)

git show: Muestra la información asociada a uno o varios objetos.

git show <nombretag>

Permite visualizar la información que recoge una tag concreta.

También puede utilizarse para extraer la información asociada a un commit indicando algunas de sus referencias, como HEAD, el SHA-1 (identificador), o incluso la rama (git show HEAD, git show 337c50f, o git show master).

Comandos sobre el repositorio local (XVI)

git merge: Realiza la integración de una rama en otra.

git merge <nombrerama> Hace un merge de <nombrerama> sobre la rama actual.

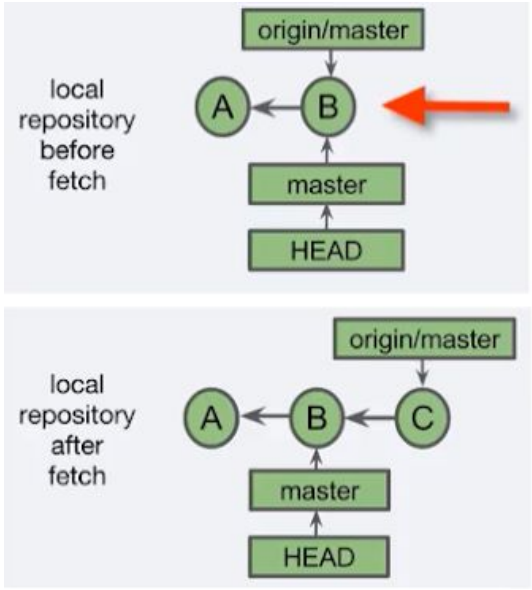
git merge --no-ff <nombrerama> Hace un merge sobre la rama actual, pero indicando que genere un merge commit incluso en el caso de poder realizarse un fast-forward (en algunos equipos de trabajo se imponen esta regla, quedando constancia del merge en un commit independiente).

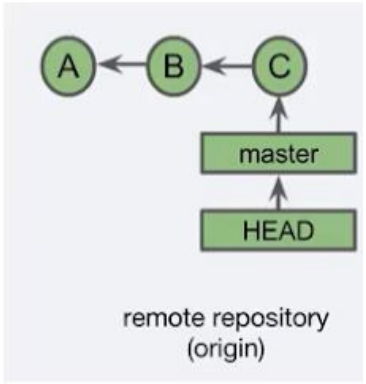
Comandos sobre el repositorio remoto (I)

git clone url: crea un repositorio local clon del repositorio remoto

Comandos sobre el repositorio remoto (II)

git fetch url: Permite descargar y visualizar los cambios de repositorio remoto sin realizar un merge con nuestro trabajo en local.

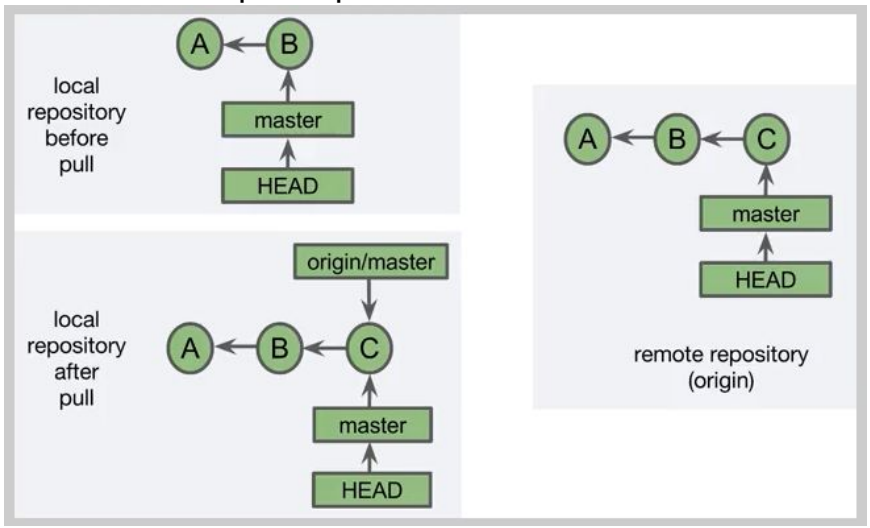




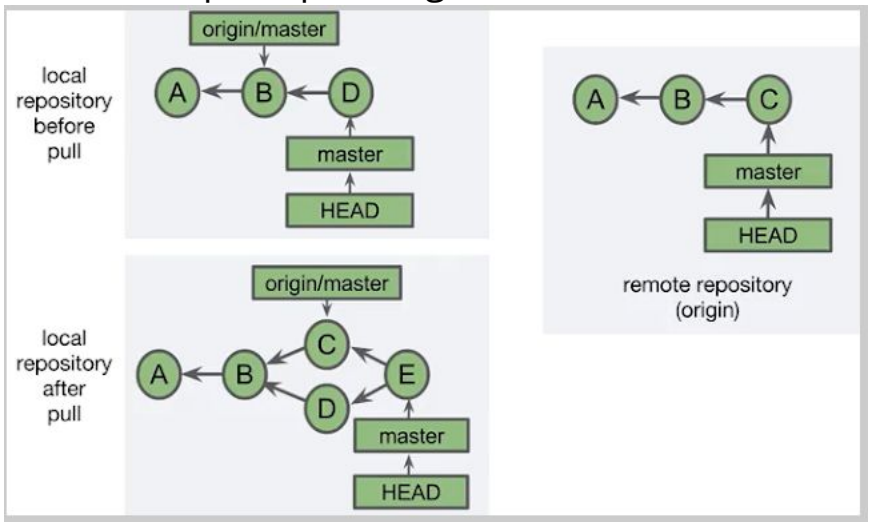
Comandos sobre el repositorio remoto (III)

git pull: combina un git fetch con git merge de la rama tracking con la rama local.

Git pull tipo fast-forward



Git pull tipo merge commit



Comandos sobre el repositorio remoto (IV)

git push: añade contenido al repositorio remoto desde el local.

Antes de hacer un push, es aconsejable hacer un pull.