

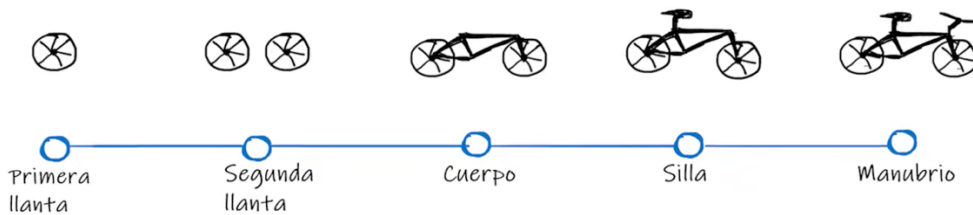


1. Introducció

- Git → és un sistema de control de versions
- Ajuda a que el desenvolupament d'un projecte sigui més fàcil de controlar.
- Consisteix en anar fent còpies (commit) que després en un futur podrem recuperar.
- També permet anar unint branques que s'han desenvolupat per separat.
- Com que git ens permet guardar canvis, desfer canvis, unir canvis, recuperar versions anteriors... es diu que git és una màquina del temps.

Exemple:

- Imaginem que estem construint una bicicleta i tenim diverses etapes:



- Git permet anar guardant còpies de cada etapa (git commit) i així en un moment futur ens permet tornar a una etapa anterior (git checkout) i així recuperar l'estat del projecte tal i com el teníem en aquell moment.

- **Conceptes importants:**

1. Historial de canvis

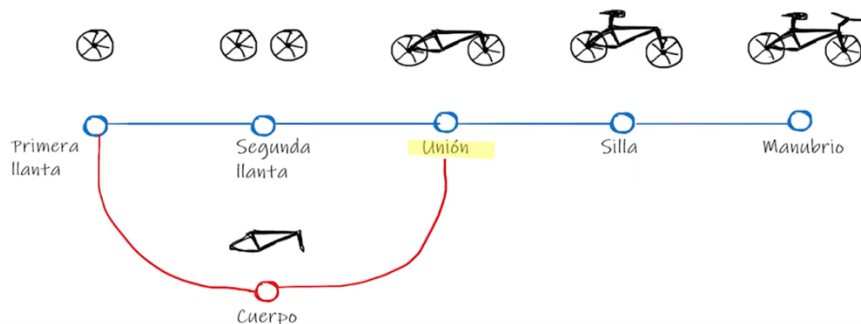
- Realment git és una màquina per a viatjar pel temps
- Cada *commit* que fem queda guardat en un historial
- En qualsevol moment podem anar enrere en el temps i seguir el projecte tal i com el teníem fa uns dies (fa uns commits)

2. Staging area

- Són totes aquelles parts del projecte (fitxers) que hem anat modificant i volem que formin part del proper *commit*
- Tot allò que no estigui a la staging area, no serà guardat en el *commit*

3. Branques

- En paral·lel 2 programadors poden anar treballant per separat en el mateix projecte, i anar fent *commits*, cadascú per la seva banda. Arribat un punt, poden ajuntar les 2 branques desenvolupades.



2. Comandes Git generals més habituals

- `git init` → permet iniciar git
- `git config` → permet configurar usuari i correu
`git config --global user.name "viladoms"`
`git config --global user.email "jv@jv.cat"`
- `git status` → permet veure un resum general del projecte (commits realitzats, dades pendents de posar a la staging area en vermell, dades posades a la staging area en verd...)
- `git add nom_fitxer` → per afegir un fitxer a la staging area
- `git commit -m "NomEtapa"` → fa commit de tots els fitxers que hi ha a la staging area
- `git log` → permet veure l'historial de commit (cada commit té un hash que el fa únic, també té un autor, data i hora de creació i comentari)
- `git log --oneline` → informació dels commits resumida (codi + nom)
- `git diff` → mostra les diferències dels fitxers modificats respecte el commit
- `git diff codiCommit1 codiCommit2` → permet veure les diferències entre 2 commits
- `git restore --staged nomFitxer` → per treure un fitxer de la staging area
- `git add .` → per afegir tots els fitxers a la staging area
- `git checkout codiCommit` → per anar a un commit anterior en el temps
- `git checkout master` → tornar al commit més recent (Regreso al futuro)
- `git restore nom_fitxer` → desfer tots els canvis que li he fet a un fitxer des de l'últim commit (recupera el fitxer com estava en l'últim commit)
- `git restore .` → desfer els canvis en tots els fitxers des de l'últim commit

3. Comandes git relacionades amb les branques

- `git branch` → mostra les branques existents del projecte
- `git branch branca2` → per a crear una nova branca en el projecte
- `git switch branca2` → per a saltar la branca 2
- `git merge branca2` → per a unir a la branca actual el que hi ha a branca2

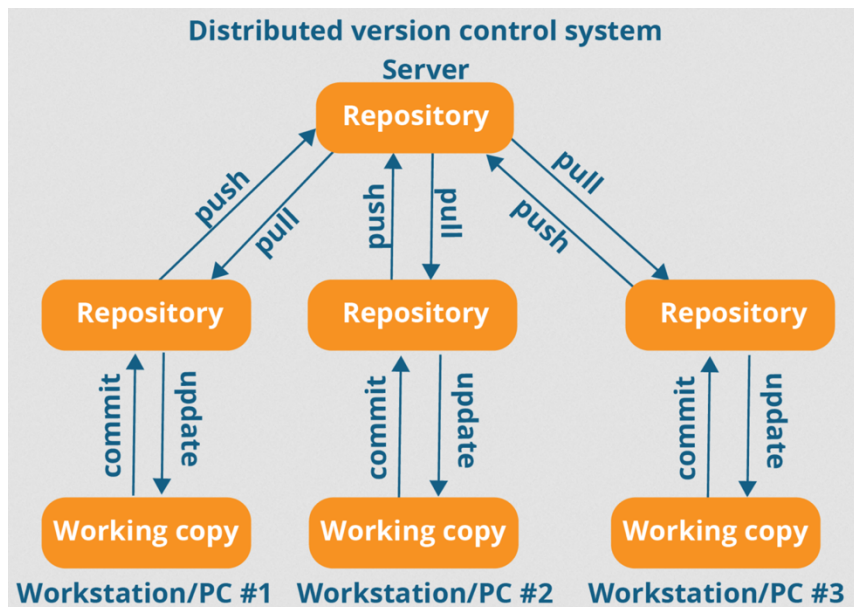
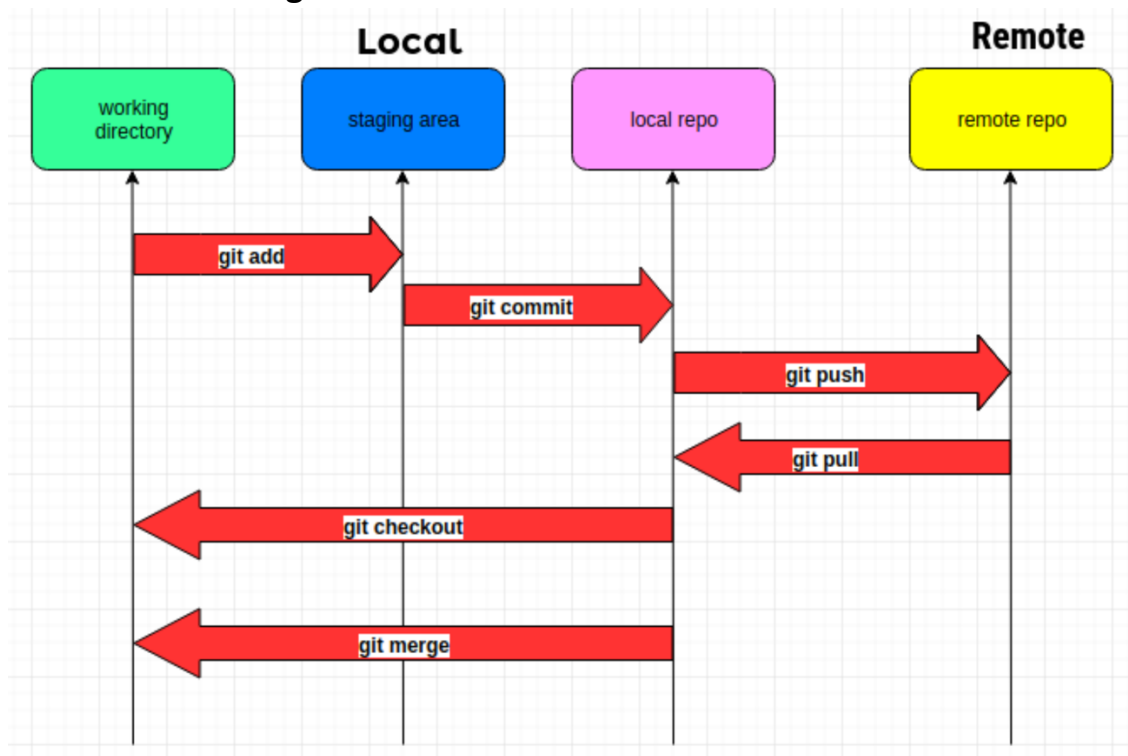
4. Comandes per a actualitzar repositori del núvol i a la inversa

- `git push` → per a pujar els canvis del repositori local al repositori del núvol
`git push https://numeroToken@github.com/nomUsuari/nomRepositori.git`
- `git pull` → per a recuperar al repositori local l'estat del repositori del núvol (habitualment origin és el remot i master és el local)
`git pull origin master`

5. Comanda per a descarregar un projecte d'un repositori remot

- `git clone http:\\adreça_repositori`

Resum comandes git



In case of fire



1. git commit
2. git push
3. leave building