Varie Terminologie

- *Repository*: set di commit, branch e tag
- *Working copy*: set di file tracciati nella copia locale
- *Commit*: un'istantanea del repository in un determinato momento.
- *Branch*: una linea di sviluppo. È un insieme ordinato di commit.
- *Merge*: l'azione di fondere due o più branch.
- *Tag*: un'etichetta personalizzata allegata a un commit.
- *Fork*: un nuovo repository che è una copia di uno esistente.
- *Pull/Merge request*: una richiesta di unire il codice da un fork o branch al repository/branch padre.

COMMIT

Il commit ha diversi campi tra cui:

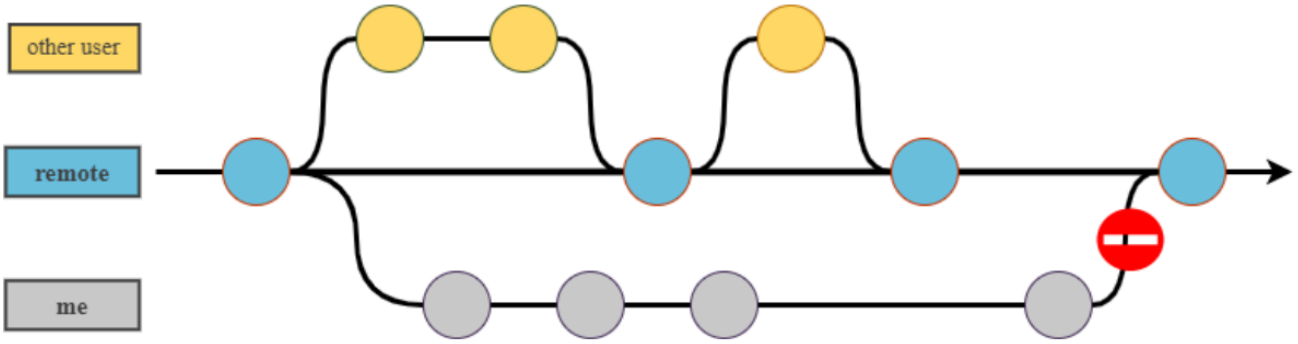
- `data+autore, data+committer`
- Commento obbligatorio
- tree: hash di tutti (~) i file nel commit
  Un ***TAG*** è un etichetta per un commit, come ad esempio le versioni, ed è meglio non cancellarli.

STAGING

Il commit può contenere un sottoinsieme delle modifiche, oppure un sottoinsieme di modifiche ad un singolo file.
Assumendo di fare commit solo dei file interi, è necessario aggiungere alla **staging area** i cambiamenti.

BRANCH

Il branch è la linea di sviluppo in cui è presente un set ordinato di commit collegati in un DAG.
Esso punta sempre all'ultimo commit ed inizia dall'ultimo orfano (solitamente il primo commit del repo). Il primo commit crea il primo branch.
Permette di lavorare in parallelo a più versioni.



HEAD

Può puntare ad un commit, branch o tag. Se non punta ad un branch:
`detached head-> no commit`

LE OPERAZIONI

IL MERGE

Fonde i cambiamenti tra due branch, se si fa merge verso un branch la destinazione contiene tutti e due i cambiamenti e l'origine rimane immutata.
Esistono 3 strategie di merge:

- ***fast-forward:*** Assumiamo di fare merge di B in A, E' valido nel caso in cui B è una continuazione di A.
- ***merge commit:*** crea un nuovo commit con due genitori ed A punta al nuovo commit
- ***rebase:*** revisionismo storico, ossia modifica la storia in modo che sia lineare ed applica il fast-forward. Ricrea ogni commit non in comune tra A e B dopo l'ultimo in A. I commit originali nel branch, ora spostato, rimangono appesi (dangling).

COMANDI ESSENZIALI

- `git init`: Inizializza un repository.
- `git status`: Visualizza lo stato.
- `git add`: Aggiunge al staging.
- `git commit` -m: Crea un commit.
- `git log`: Visualizza la storia.

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
git init
git status
touch test.txt
git add test.txt
git status
mkdir thedir
mv test.txt thedir/
git add thedir/
git commit -m 'made a dir'
```