

# INFORMAZIONI SUL CORSO

Email:

`cuteri@mat.unical.it`

Ricevimento:

Martedì alle 15:00

# GIT

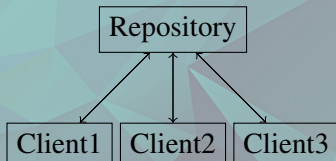
Bernardo Cuteri



**git**

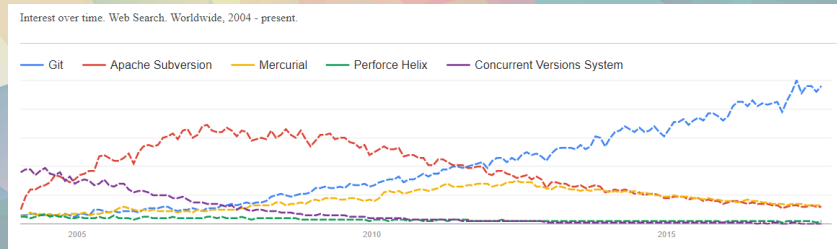
# Cos'È GIT?

- Git is a software per il controllo delle versione distribuito
- Permette a gruppi di sviluppatori (client) di lavorare simultaneamente a più file
- I file sono conservati in un repository online condiviso, e i client lavorano su copie locali (dette anche repository locali)
- Git è basato su un sistema a branch (rami in italiano)



# ALTRI SOFTWARE DI VERSIONING?

Confronto di popolarità tra Git e gli altri sistemi più utilizzati per il controllo della versione software.



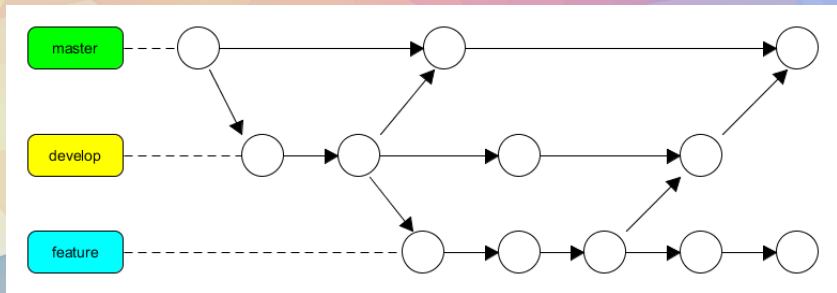
# COSA SI PUÒ FARE CON GIT?

Alcune operazioni che i client possono effettuare con Git:

- Creare una copia locale dei file memorizzati sul repository online
- Modificare la propria copia locale offline
- Committare i cambiamenti unitamente ad un messaggio che specifica cosa è stato cambiato
- Effettuare un push delle proprie modifiche e renderle disponibili nel repository centrale

I commit sono come delle foto (snapshot) del codice, accompagnate da un messaggio. Si può sempre tornare ad un commit precedente e gli utenti possono selezionare quali file committare e quali escludere.

# GIT BRANCHING



- Nell'immagine ogni riga rappresenta un branch
- I cerchi rappresentano invece i commit
- Un nuovo branch viene creato sempre come copia di un branch pre-esistente
- Le frecce che si spostano indietro su un branch rappresentano i cosiddetti *merge*
- Il branch *master* è il branch iniziale di default di Git.

# GIT BRANCHING

- Ci si può spostare da un branch all'altro in qualsiasi momento

Quando si realizza una nuova feature si può (non è obbligatorio):

- Creare un nuovo branch
- Aggiungere la feature in uno o più commit
- Unire il branch nuovo al branch di partenza per aggiungere la feature

Questo ci permette di lavorare in isolamento (esempio: rispetto ad altre feature) e di poter interrompere i lavori in corso d'opera e tornarci più tardi (esempio, c'è un bug urgente da risolvere sul master).

I branch possono essere:

- Remoti
- Locali

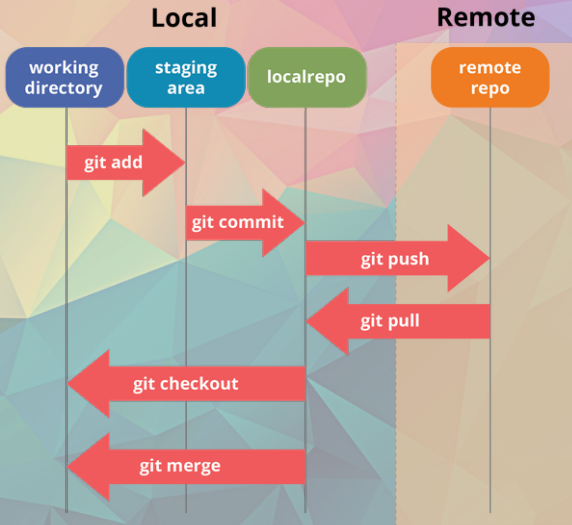
I client lavorano sui branch locali e possono sincronizzare i branch locali con quelli remoti, che invece rappresentano una copia dei branch che si trovano sul repository centrale.

# COMANDI E FUNZIONAMENTO

## Comandi essenziali:

- `git pull`
  - Aggiorna i branch locali con le eventuali modifiche presenti sul repository remoto
- `git add`
  - Aggiunge un file al repository locale (permettendone la condivisione)
- `git commit`
  - Committa i cambiamenti apportati alla copia locale (e aggiunti nella cosiddetta area di staging)
- `git push`
  - Carica i cambiamenti locali committati sul repository remoto.  
Nota: se qualcun altro ha apportato delle modifiche al repository remoto dopo il nostro ultimo aggiornamento (pull), allora, prima di poter effettuare il caricamento (push) siamo costretti a effettuare un ulteriore pull il quale unirà le modifiche locali con quelle remote.





- git status
  - Ci informa sullo stato del repository locale: quali file abbiamo aggiunto, modificato, quali commit sono stati effettuati, su quale branch ci troviamo...
- git branch
  - Tramite opportune opzioni può essere utilizzato per sapere quali sono i branch, aggiungerne di nuovi o cancellarne qualcuno
- git checkout *nome*
  - Permette di spostarci su un altro branch o ad uno specifico commit. Può anche essere utilizzato per creare un nuovo branch e spostarci con l'opzione -b.

- Ci sono molti altri comandi!
- git stash
- git merge
- git fetch
- ...
- Date un'occhiata quì:  
<https://www.youtube.com/watch?v=ZDR433b0HJY> per avere  
una guida completa (in inglese)

- Creiamo un nuovo account su un servizio di hosting Git gratuito:
  - <https://github.com/>