

# VCS-GIT

## 3-Social coding

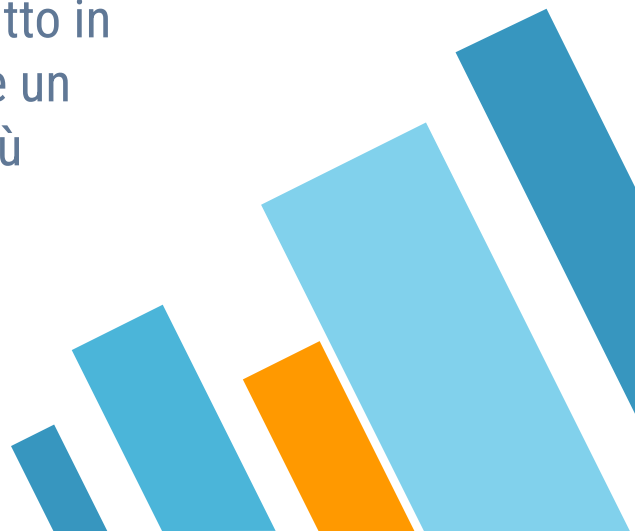
Prof. Marcello Missiroli





## DI CHE SI PARLA?

Avete visto i primi usi di Git in locale. Già così fornisce una forma di “backup specializzato” per il vostro progetto. Però la sua potenza si dimostra soprattutto in congiunzione a un repository esterno, che fornisce un backup aggiuntivo e la possibilità di lavorare da più stazioni di lavoro e con più persone.





**GitHub**



**git**

**VS**

Atlassian  
 **Bitbucket**



**GitLab**





1.

**Coding goes social**



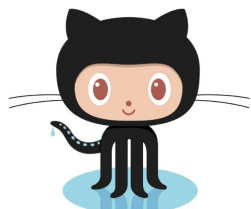
## Le origini

L'arrivo di Internet 2.0, la necessità di condividere il codice con la comunità, il bisogno di accedere al codice sempre, comunque e da dovunque ha portato alla nascita dei primi siti di condivisione codice, come ad esempio Sourceforge (1999).

Alcuni di questi siti, grazie anche alla potenzialità di networking di Git, si sono evoluti in vere e proprie comunità di social networking




# Github



# Git**H**ub

Fondato nel 2008, si tratta del sito sicuramente più famoso e con maggior numero di utenti. Fu comprato nel 2018 da Microsoft causando sconcerto e preoccupazione soprattutto nella comunità FLOSS.

Ciononostante, il sito è florido e ben congegnato. È tuttora la scelta principale per i progetti open source, grazie alla sua lunga storia e alle funzionalità di ricerca interna.





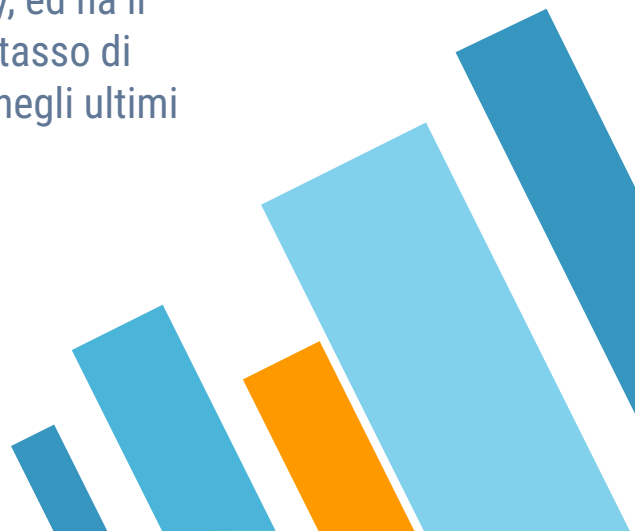
# Gitlab



# GitLab

Lanciato nel 2014 dalla due sviluppatori ucraini, il sistema è cresciuto considerevolmente negli anni e si distingue dai prodotti simili per essere open source ed essere ottimizzato per il modello di sviluppo DevOps.

È favorito dalle aziende con problemi di privacy, ed ha il maggior tasso di crescita negli ultimi anni.



# Bitbucket



Il servizio è di proprietà di Atlassian, un grosso calibro nella gestione automatica dei progetti, noto in particolare per la suite Jira. Per questo motivo, è particolarmente diffuso nelle aziende medio-grandi.

Suo punto di forza è l'integrazione con altri servizi come JIRA Software, HipChat, Confluence e Bamboo.



## Su che basi scegliere?

**Github** ha il miglior supporto della community, essendo il più anziano e pullula di progetti interessanti. Pende l'ombra di Microsoft.

**Gitlab** permette di lavorare installando un server privato e si concentra sulla catena di produzione

**BitBucket** è sicuramente la scelta più “corporate”.

## E quindi?

Per i nostri interessi, tutte queste aziende offrono la possibilità di utilizzare gratuitamente i loro servizi, e regalano non meno di 500Gb per ogni repository. Le differenze s'iniziano a vedere a partire dai piani a pagamento. Qui faremo riferimento soprattutto a **Gitlab**, per via del suo impegno a favore del software open source.

È molto probabile che li userete tutti e tre.



“I’m an egotistical bastard,  
and I name all my products  
after myself. First Linux, then  
Git.”

*Linus Torvalds.*

Git: a foolish or  
worthless  
person.

# 2.

## Git remotes

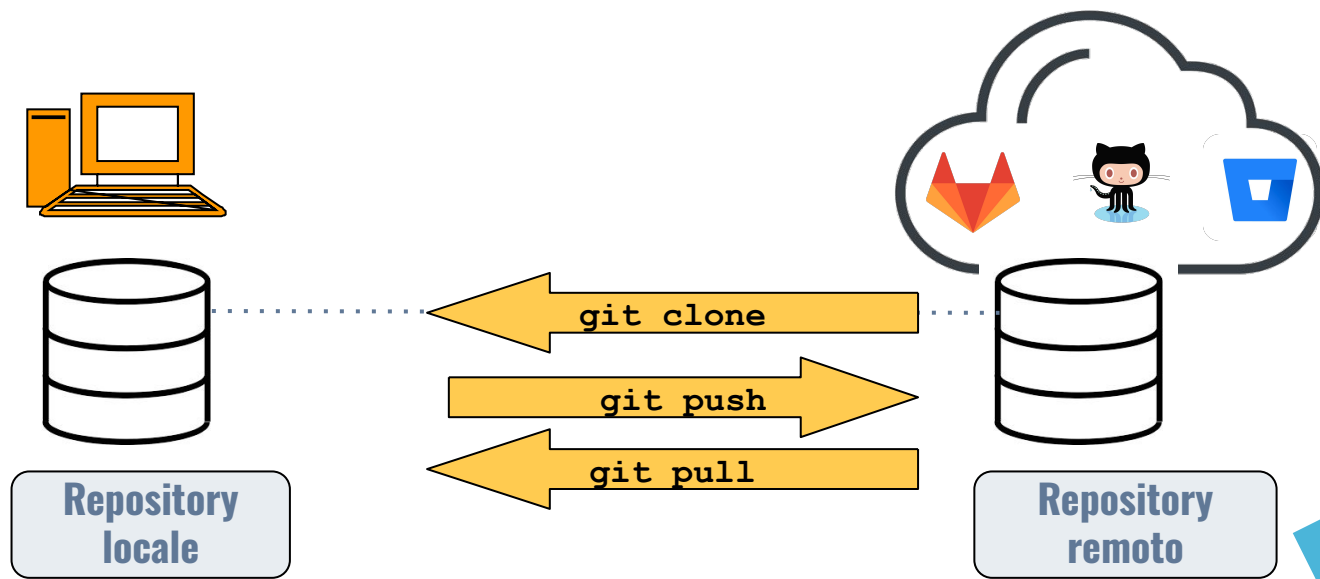
Backup senza sforzo

## E quindi?

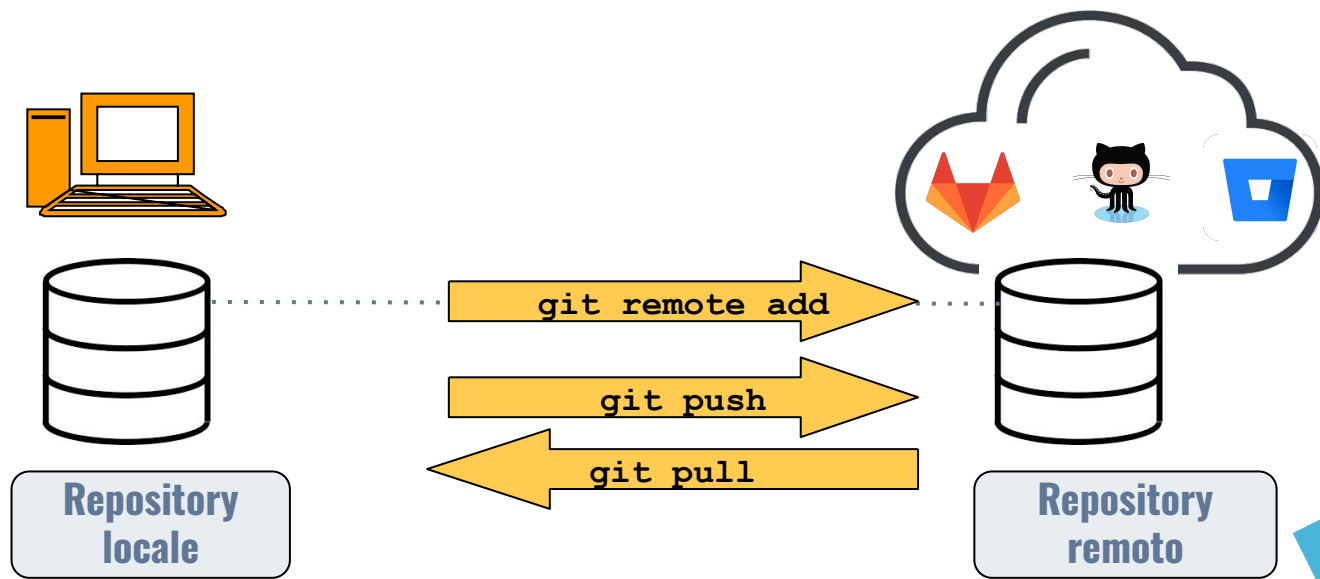
Per git, tutti i repository sono uguali, e il fatto che si possa accedere con molti protocolli (ssh, http, copia locale) facilita la loro interazione - è per questo che è definito come DVCS, dove **D** sta per Distribuito.

Esistono però comandi specifici per interagire con altri repository (in gergo: “remotes”) in modo da velocizzare il lavoro

# Schema di base



## Schema alternativo



## Torniamo a Bob

Ora, sappiamo che Bob ha già un repo locale, per cui rientra nel secondo caso.

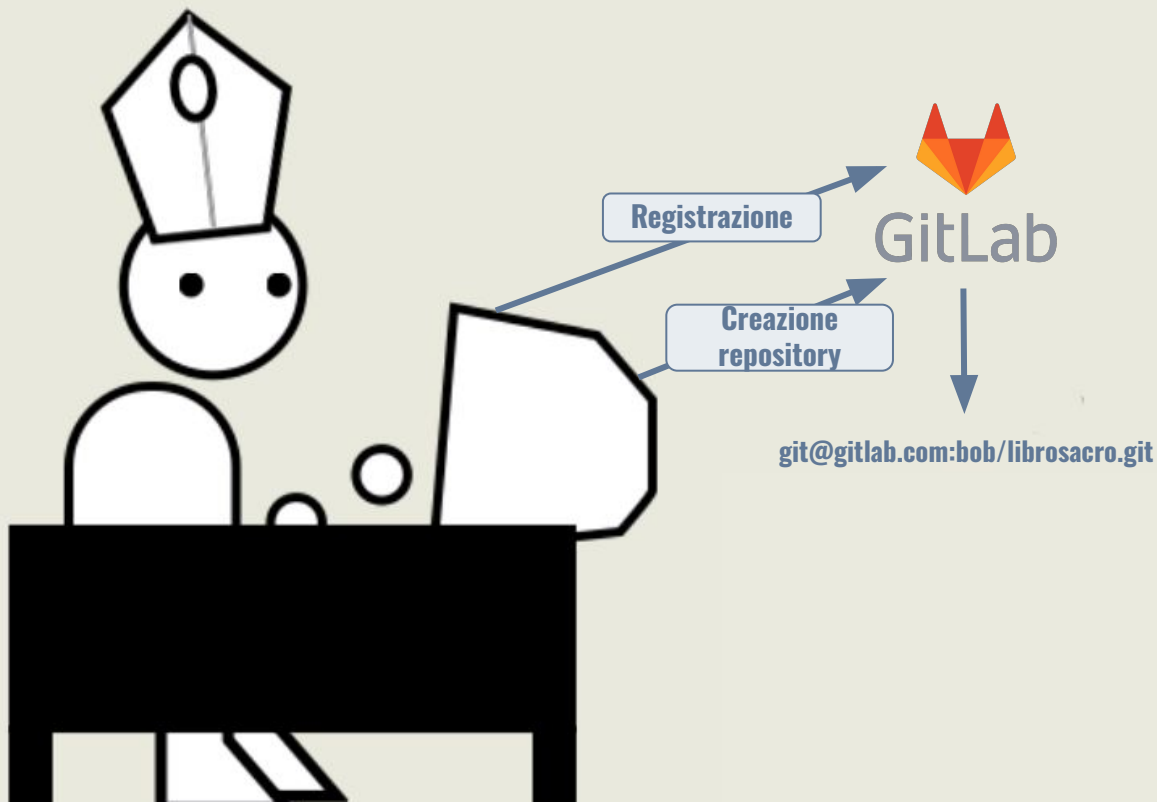
Decide di usare Gitlab come piattaforma per condividere il suo pensiero.

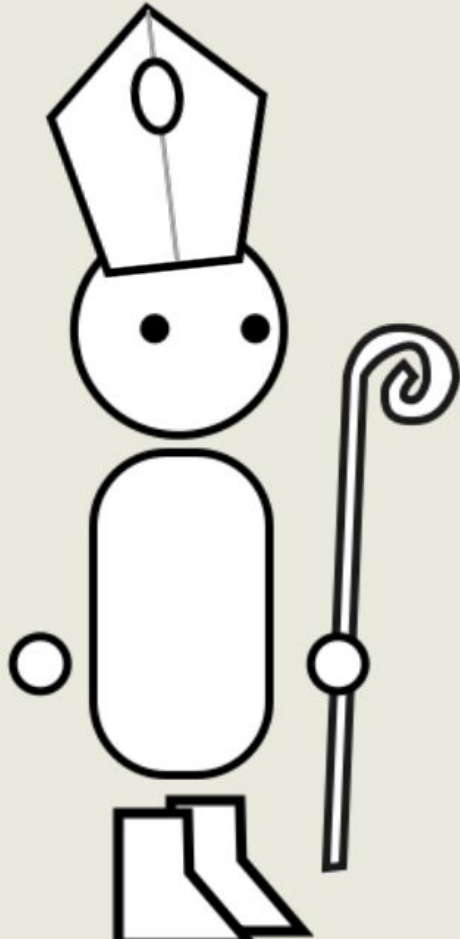


# Unimore PdS stops here

Passiamo al pratico

# Il libro va in rete





```
$ git remote add origin  
git@gitlab.com:bob/libronesac  
ro.git
```

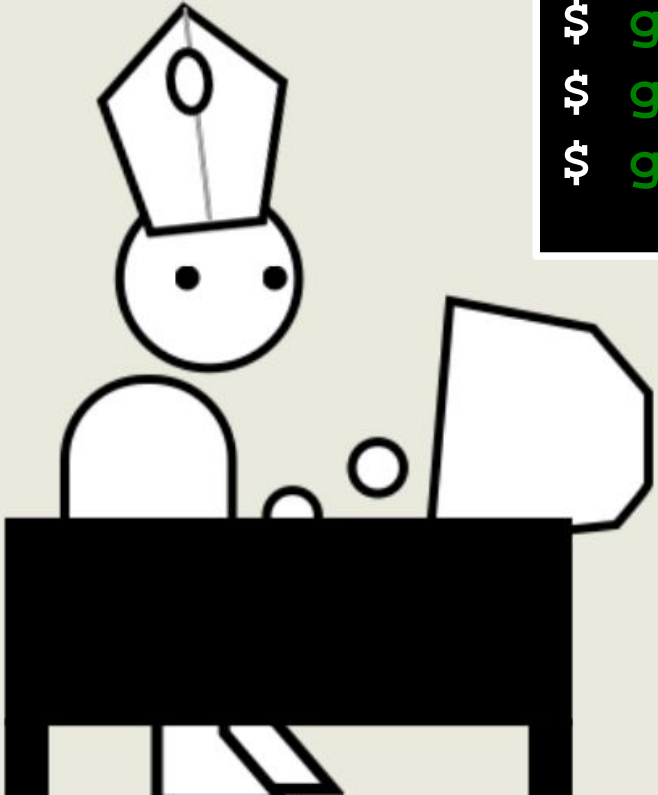
```
$git push -u origin --all  
--force
```

```
[Password?]
```

# Prosecuzione del lavoro

Ora bob può proseguire nel lavoro facendo commit localmente, e ogni tanto (a fine giornata?) fa un push per salvare lo stato dei lavori

```
$ git commit -m 'Appendice'  
$ git commit -m 'Piccolo errore'  
$ git commit -m 'Riflessione'  
$ git push
```





# 3.

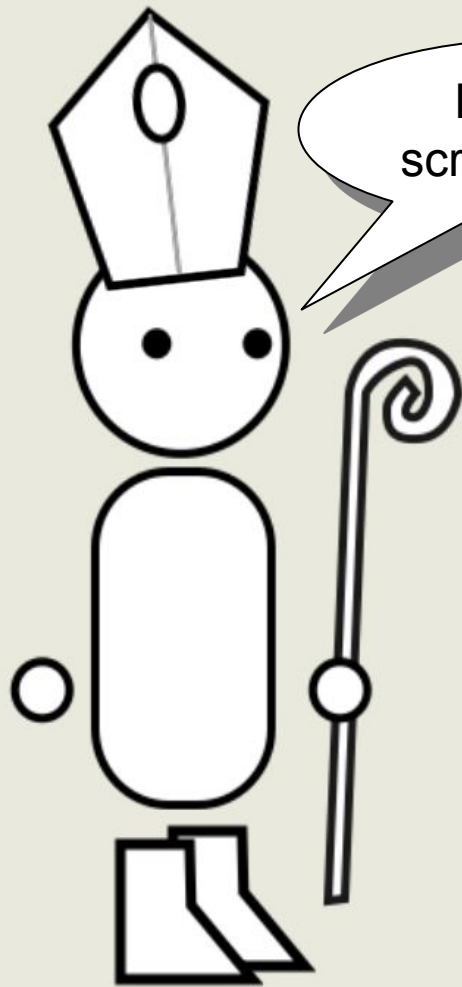
## **Social, davvero**

Aggiungiamo i collaboratori



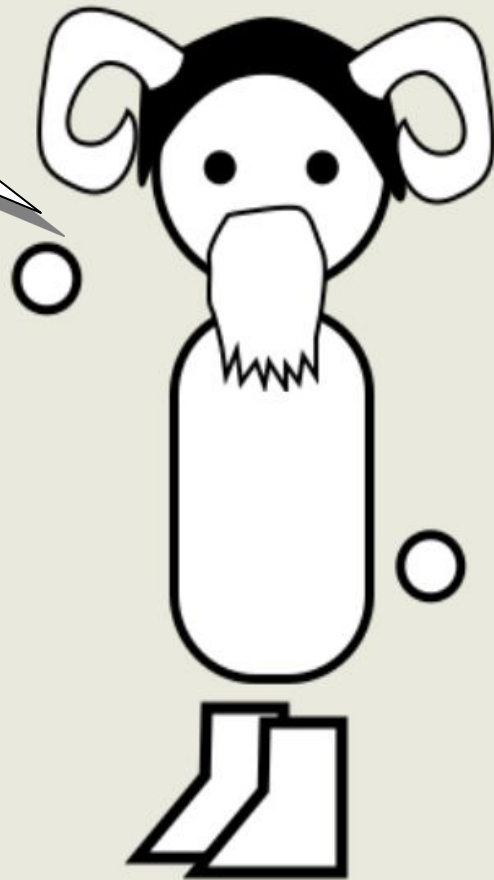
## **Bob ha fatto proseliti.**

Tim si è unito alla chiesa di Bob e ha deciso di aiutarlo.  
Dato che non si trova nella stessa città di Bob, dovrà utilizzare la rete per sincronizzarsi nello sviluppo.



Insieme  
scriveremo...

Ciao, sono  
Tim



**Il Librone sacro!**



## La chiesa di Bob

Dopo aver cambiato il nome al repository, Tim deve scaricare una copia di lavoro. Tim si trova quindi nella prima situazione e per lavorare deve clonare localmente il repo (dopo essere stato autorizzato da Bob, ), quindi inizia a lavorare. Bob può continuare a lavorare in parallelo.

```
$ git clone  
git@gitlab.com:bob/libro  
nesacro.git  
  
$ cd libronesacro
```



```
$ git add 'capitolo 5.txt'  
$ git commit -m 'Quinto  
capitolo'  
$ git push origin master
```



```
$ git pull origin master  
$ ls  
capitolo 1.txt  capitolo 2.txt  
capitolo 3.txt  capitolo 4.txt  
capitolo 5.txt  capitolo 5.txt
```

# 4.


## Conflitti

Collaboratori == problemi?



## Chi ha ragione?

Finora Tim e Bob hanno lavorato su file diversi, per cui non ci sono stati problemi. Ma che accade se lavorassero sugli stessi file?



# Conflitto

```
$ git pull origin master
$ git merge comandamenti
Auto-merging Capitolo 1.txt
CONFLICT (content):
Merge conflict in Capitolo
1.txt
Automatic merge failed;
  fix conflicts and then
commit the result.
```

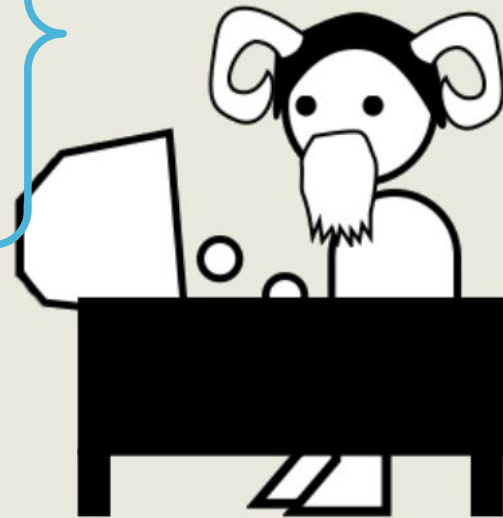


# Come appare il conflitto

Versione presente  
sul server remoto

```
<<<<<< HEAD:Capitolo 1.txt  
All'inizio ci fu Bob, che  
=====  
All'inizio ci fu Tim, che  
>>>>>> 364d070c8e
```

Versione locale on commit ID



# Come risolvere il conflitto

All'inizio ci fu Bob, che

Il supremo Tim  
ha sempre  
ragione...

Save &  
commit!





# Rivediamo qualche concetto

## Remote

Un repository presente altrove, accessibile via rete

## Origin

Nome convenzionale del repo remoto

## Push

Commit verso il server remoto

## Pull

Aggiornamento dal server

## Clone

Copiatura locale iniziale di un repo remoto

## Conflitto

Quando la versione presente non può essere fusa automaticamente con quella precedente.



# **GRAZIE!**

**...e siamo a metà  
dell'opera....**



# CREDITS

Ringraziamenti a :

- » Presentation template by SlidesCarnival
- » Photographs by Unsplash
- » Anil Gupta ([www.guptaanil.com](http://www.guptaanil.com))
- » Pete Nicholls ([github.com/Aupajo](https://github.com/Aupajo))
- » Armando Fox

***Questo documento è distribuito con  
licenza CreativeCommon BY-SA 3.0***