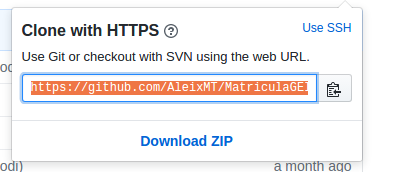
**Configuració global d’usuari**

git config –-global user.name “AleixMT”

git config –-global user.email [aleixaretra@gmail.com](mailto:aleixaretra@gmail.com)

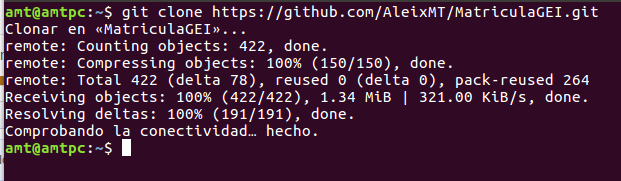
Això serveix per a configurar el teu usuari global de git. Per a servidor de github no cal fer-ho.

**git clone**

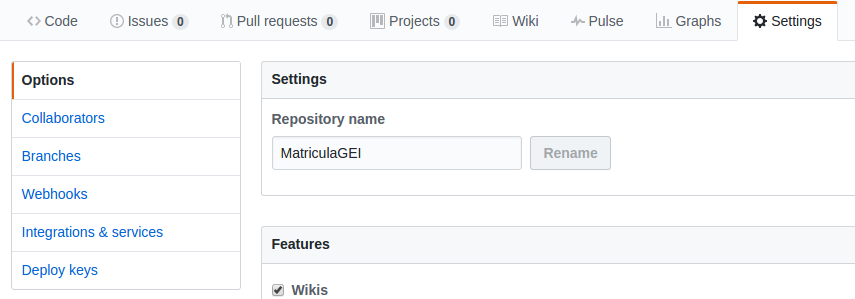
Serveix per a crear el repositori de nou a l’ordinador local (el teu propi PC). El git es baixarà tot el contingut del repositori a partir d’on apunta el teu punter de rama (això no estic segur però es pot modificar després). Sense fer el git clone abans no pots entrar a treballar al teu repositori.

Es pot fer clone a partir del protocol ssh o del protocol http. El ssh a mi mai em funciona (crec que has de fer alguna config. Adicional). El més fàcil es utilitzar http. La direcció la proporciona git al crear un nou repositori (a les instruccions per a inicialitzar-lo) o bé aquí (imatge):

git clone <Direcció.http>



**Afegir colaboradors**



Github és públic, de manera que tothom pot baixar-se el teu codi i treballar amb ell. El que no poden fer és fer push (tenen permís de lectura però no d’escriptura). Per a donar permisos a algú, has de fer settings → collaborators i afegir l’usuari de github a qui li donis permís. Necessites la contrassenya.

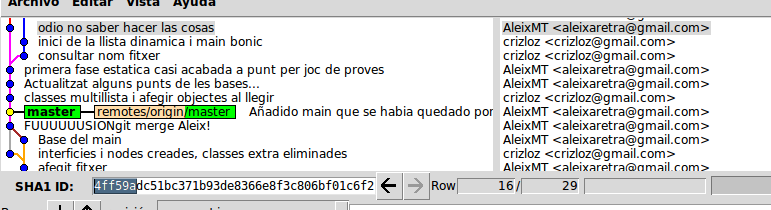
**Gitk**

És una aplicació gràfica per a veure els commits en un repositori local.

invoca’l de la següent manera:

gitk –-all –-date-order &

Mostra tots els commits ordenats perdata en segon pla.



- Per a actualitzar el que veus: Archivo → reload (per si fas per exemple un fetch quan git ja està obert)

- Allò que posa SHA1 ID serveix per a referenciar commits. Very importante mai frend.

- Allò que posa remotes/origin/master fa referència a la versió que hi ha en el servidor (és una mica random). La idea es que el remotes/origin/master i la etiqueta master haurien d’estar al mateix lloc. Encara que no té per que

**git fetch**

Actualitza el que tu veus del servidor local. Molt important fer-ho abans de fer qualsevol cosa (especialment un push), ja que pots estar fent push a partir d’un commmit que no existeix i deixar-lo sense penjar de res. Vamos que fetch abans de fer qualsevol cosa. Fes tants fetchs com cops agafis aire.

git fetch

**git pull**

Jo no l’utilitzo gaire perque sóc més de modificar els apuntadors de ramas manualment. Pero git pull el que fa és baixar-te l’últim commit al que apunta la teva rama i actualitzar el remote/origin. Fes-ho servir junt amb git branch -f (aquí baix).

**git branch**

serveix per a crear una nova branca al teu repositori. La branca eria com un usuari, però no noes més que un apuntador a un commit. Lo habitual es tenir tantes branques com colaboradors + 1(el master).

Git branch nombranca

Per a canviar un apuntador de rama. (AQUESTA COMANDA ES SUPER IMPORTANT PER A CONTROLAR COMPLETAMENT EL GIT).

Git branch -f nombranca SHAIDcommit



fem

git branch -f Aleix <SHAIDcommit al que apunta Kros (gem de consultar-lo abans)>

fem un reload a gitk i…



Hem mogut l’apuntador de rama.

**git checkout**

Serveix per a “chequear” un determinat commit. És a dir, per a tornar enrere i veure com era per si has fet un commit que no tocava. Has d’utilitzar el id que et proporciona gitk, o el que et va sortint a la pantalla cada cop que fas alguna operació gran sobre el servidor.

Git checkout <SHA1ID>

A partir d’aquí pots fer commits i tal i continuar per unn altre lloc, es a dir, el commit divergiria en dos, el commit que no funciona que es queda abandonat (dangling commit) i el commit per on has seguit treballant. Els danging es borren amb el temps si no m’equivoco.

Git checkout -b nombranca

// és el mateix que

git branch nom

git checkout nom

Crea una nova branca al lloc o apunta el commit actual i entra en ella.

Git checkout nombranca

Serveix per a cambiar a una brancaque ja existeix.

**Git reset**

A vegades vols fer checkout o fer un pull o alguna cosa i et dira això o algo semblant

error: Your local changes to the following files would be overwritten by merge:

wp-content/w3tc-config/master.php

Please, commit your changes or stash them before you can merge.

Això vol dir que hi han canvis i per tant no pots fer el que volies. Normalment això passa quan tens un canvi en plan un espai que has posat al llegir el document i e git et fa la vaina. Per a solucionar això has de fer

git reset –hard

Això t’actualitza a la versió del commit on apunta la teva rama, i per tant desapreixeran aquests canvis no desitjats.

**Git status**

mostra informació…

**git push**

Serveix per a pujar commits. Es pot fer simplement teclejant git push. Git push puja fitxers, de vegades, si simplement has fet un git branch -f, has d’actualitzar les refereències, pel que has de fer

git push –-all

(años de experiencia).

Git push origin nomrama

Actualitza el servidor local i el remot

**git add**

Afegeix nous arxius al seguiment. Quan creis nous arxius (crear una nova clase en java, un fitxer de text etc.) has de fer

git add nomfitxer

El que jo faig es

git add -A

que afegeix tots els fitxers de forma recursiva…