

Instal·lació i configuració bàsica

Git ja ve instal·lat per defecte en la gran majoria de distribucions linux.

En un debian, per exemple, es pot instal·lar així:

```
$ sudo apt install git
```

Abans de començar a treballar amb git, n'hem de fer una configuració mínima:

- Configuració global: serveix per a tots els repositoris que creem.
- Configuració pròpia per a cada repositori.

Usarem una configuració global, que serà la que tindrem per defecte i que s'aplicarà a tots els repositoris que creem. Això no impedeix crear una configuració específica per a un repositori en particular si així ho necessitem.

Cal configurar com a mínim els següents dos paràmetres: el nom, i l'adreça de correu electrònic. Les següents comandes es poden executar des de qualsevol ubicació dins del sistema:

```
$ git config --global user.name "Jordi Casamitjana Querol"  
$ git config --global user.email "jcasamq@domini.cat"
```

Les comandes per a fer una configuració específica per a un repositori en concret són les mateixes, però sense l'opció `--global`. En aquest cas, serà necessari executar-les dins del repositori.

Si ja tenim un compte a github o gitlab, és recomanable també configurar el nom d'usuari del nostre compte de github o gitlab:

```
$ git config --global user.username "jcasamq"
```

En aquest darrer cas (si tenim un compte de github o gitlab) també configurarem l'adreça de correu i el nom perquè coincideixi amb els del nostre compte.

Ignorant fitxers: .gitignore

Dins d'un repositori git, tots els fitxers estan sotmesos a git. Però si ens interessa que algun fitxer o directori no ho estigui, el podem incloure en un fitxer especial, anomenat `.gitignore` que serveix precisament per a això: tot fitxer o directori inclòs a `.gitignore` no serà tingut en compte per git.

És convenient no incloure el mateix fitxer `.gitignore` dins de `.gitignore`, ja que convé que git no ignori el fitxer `.gitignore`.

Podeu trobar informació sobre com es configura el fitxer `.gitignore` en el següent enllaç:

[Què és un fitxer `.gitignore` i com el puc configurar?](#)

Exercici: Creació del primer repositori

Suposem que ja tenim git instal·lat. En cas contrari, l'instal·leu segons quin sigui el vostre sistema operatiu. Anteriorment hem vist un exemple de com instal·lar-lo en una distro Debian:

```
$ sudo apt install git
```

Obriu un terminal en el vostre entorn de treball.

- Primer pas: establir la configuració global

Si ja teniu un compte a Github o Gitlab, heu de fer servir les mateixes dades que hi teniu per fer la configuració local:

```
$ git config --global user.name "Jordi Casamitjana Querol"
$ git config --global user.email "jcasamq@domini.cat"
$ git config --global user.username "jcasamq"
```

Per a comprovar la configuració que tenim, podem executar:

```
$ git config -l
user.name=Jordi Casamitjana Querol
user.email=jcasamq@domini.cat
user.username=jcasamq
```

Els valors de configuració que no hem establert nosaltres, tenen uns valors per defecte, que no caldrà canviar.

- Segon pas: establir el directori de treball

Per fer totes les proves i exercicis d'aquest curs de git, crearem un directori anomenat git en el vostre `$HOME`. Aquest directori serà el nostre directori de treball:

```
$ mkdir ~/git
$ cd ~/git
```

- Tercer pas: establir el directori matriu del repositori

En el nostre directori de treball, crearem un subdirectori anomenat `exemple1_git`:

```
~/git$ mkdir exemple1_git
~/git$ cd exemple1_git
```

Aquest nou directori està buit, doncs l'acabem de crear. Això ho podem comprovar amb la comanda `ls -a`:

```
~/git/exemple1_git$ ls -a
.  ..
~/git/exemple1_git$
```

- Quart pas: inicialitzem el repositori git

Inicialitzem aquest directori com a repositori git. Des de dins del mateix directori, hi executem `git init`. Un cop completada la comanda, si llistem el contingut del directori, ara ja hi veurem el subdirectori `.git` (que conté les metadades de git):

```
~/git/exemple1_git$ git init
Initialized empty Git repository in /home/jcasamq/git/exemple1_git/.git/
~/git/exemple1_git$ ls -a
.  ..  .git
~/git/exemple1_git$
```

- Cinquè pas: creem el fitxer `.gitignore`

A continuació hi crearem el fitxer `.gitignore`, encara que de moment no el farem servir, convé tenir-lo present des de bon inici:

```
~/git/exemple1_git$ touch .gitignore
~/git/exemple1_git$ ls -a
.  ..  .git  .gitignore
~/git/exemple1_git$
```

- Sisè pas: creem el fitxer `README.md`

Tot repositori git convé que tingui un fitxer anomenat `README.md`. Aquest és un fitxer *markdown* que té per finalitat donar informació bàsica sobre els continguts del repositori.

Un fitxer *markdown* es pot crear i editar des del terminal de linux, usant els editors `vi` o `nano` (entre d'altres) o bé es pot crear i editar usant l'entorn *Jupyter Lab*:

```
~/git/exemple1_git$ vi README.md
```

```
# El meu primer exemple de repositori git
```

```
~/git/exemple1_git$ ls -a
.  ..  .git  .gitignore  README.md
```

```
~/git/exemple1_git$
```

Si feu servir l'entorn *Jupyter Lab* abans de crear el fitxer README.md, us heu d'ubicar en el directori on el voleu crear (a ~/git/exemple1_git en aquest cas). Hi escriviu el següent contingut i guardeu els canvis (ctrl+s):

```
# El meu primer exemple de repositori git
```

- Setè pas: comprovem l'estat del nostre repositori

Amb la comanda `git status` obtenim informació sobre en quin estat es troba el nostre repositori:

```
~/git/exemple1_git$ git status
```

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
.gitignore
```

```
README.md
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
~/git/exemple1_git$
```

En properes activitats aprendrem a entendre el significat d'aquest missatge.