

Activitat 1: Git

Introducció

En aquesta activitat aprendrem a utilitzar les bases de Git i a fer-lo servir des de la línia de comandaments que és com se li pot treure tot el profit a aquesta eina.

Per a fer aquesta activitat es recomana l'ús de Ubuntu LTS (p.e. 22.04).
Hauràs de fer captures de cada pas i retallar-les per tal que es vegi bé en aquest document que hauràs de lliurar de l'activitat.

NOTA: per comprovar que alguna comanda ha canviat correctament l'estat del git farem «git status»

Primers passos

1. Instal·la git en el Linux. Si utilitzes Ubuntu
2. Configura el nom i el correu electrònic
3. Com inicialitzaries una carpeta per a que fora un repositori de git? Crea una carpeta activitat_git i inicialitza git dins (pista: init)

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git
$ echo "# activitat_git" >> README.md

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git
$ git init
Initialized empty Git repository in C:/Users/mrodr/Desktop/JaumeBalmes/M8/UF4/activitat_git/.git/

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (master)
$ git commit -m "First commit"
[master (root-commit) f9fd07c] First commit
1 file changed, 1 insertion(+)
 create mode 100644 README.md

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (master)
$ git branch -M main

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git remote add origin https://github.com/mrodriguezzsx/activitat_git.git

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 232 bytes | 116.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mrodriguezzsx/activitat_git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$
```

4. Crea un arxiu index.html amb un contingut qualsevol de html que sigui visible en el navegador. En quin estat es troba respecte al Git?

En aquest cas, esta untraked

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html

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

5. Què faries per tal d'afegir el fitxer index.html en el control de versions?→ fes-ho.
git add index.html

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git add index.html
```

6. Crea els arxius code.js i styles.css a la carpeta activitat_git. Si ara tens diversos arxius nous com faries per afegir-los tots a l'hora amb una sola comanda? → fes-ho
git add .

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git add .
```

7. En quin estat està un fitxer que s'ha afegit al git? Ja té els canvis com a definitius en el repositori? Què és la «staging area» de git? Amb quina comanda fas permanents els canvis al repositori des de la staging area? → fes-ho






Estan en estat Staging.

Els canvis encara no estan al repository.

Per a fer permanents els canvis s'ha de fer un git commit i un git push.

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git commit -m "2n push"
[main 535612e] 2n push
3 files changed, 1 insertion(+)
create mode 100644 index.html
create mode 100644 main.js
create mode 100644 styles.css

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 347 bytes | 347.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mrodriguezzsx/activitat_git.git
f9fd07c..535612e main -> main
```

	mrodriguezzsx 2n push	
	README.md	First commit
	index.html	2n push
	main.js	2n push
	styles.css	2n push

Avançant amb git

8. Afegeix un arxiu a la staging area. → mostra-ho. Com el trauries del stage? → fes-ho

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git add prova.txt

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git res
reset      restore

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git restore
.git/      README.md  index.html  main.js     prova.txt   styles.css

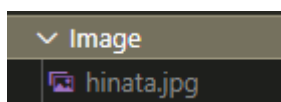
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git restore prova.txt
```

9. Fes canvis en un arxiu (pe. code.js) i guarda'ls (status). Com faries ara per descartar els canvis i tornar a la versió anterior? (pista checkout) → fes-ho
10. Si crees arxius que no vols mantenir dins del git com compilats o imatges, ... com faries per tal d'excloure'ls del git i que els ignore? Crea una carpeta images i fica dins una imatge. Fes que sigui ignorada per el git (pista .gitignore)

```

♦ .gitignore
1  #Arxius per ignorar:
2
3  *.png
4  *.jpg
5  *.gif

```



Es veu que a la imatge no està agafant el .jpg per pujar-ho al GitHub

11. Suposem que esborres un arxiu del projecte per error (fes-ho). Com faries que git te'l restaurés? → fes-ho

```

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBaimes/M8/UF4/activitat_git (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    index.html

no changes added to commit (use "git add" and/or "git commit -a")

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBaimes/M8/UF4/activitat_git (main)
$ git restore index.html

```

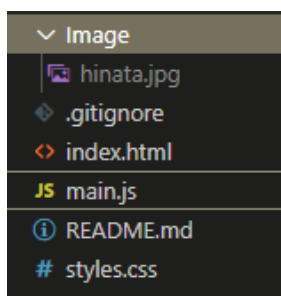
12. I si ara vols esborrar de veritat un arxiu del projecte... com ho faries amb git? → fes-ho i fes commit

```

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBaimes/M8/UF4/activitat_git (main)
$ git rm prova.txt
rm 'prova.txt'

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBaimes/M8/UF4/activitat_git (main)
$ git commit -m "Prova.txt eliminada"
[main 323051c] Prova.txt eliminada
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 prova.txt

```



13. Suposem que vols tornar en el temps a l'estat d'un commit anterior per recuperar l'arxiu esborrat... com ho faries (pista ajuda't amb log) fes-ho

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git checkout 03962658ea5e92681c020616dbfecf06b8b323da
Note: switching to '03962658ea5e92681c020616dbfecf06b8b323da'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 0396265 Git Ignore
```

14. Suposem que ara volem iniciar un grup de canvis que no volem que es barregen amb el que treballa la resta del grup. Hem de crear una branca. Com crearem una branca anomenada la_meva_branca? → fes-ho

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git ((0396265...))
$ git branch matias
```

15. Si ara decidim abandonar la línia de desenvolupament la_meva_branca i volem tornar a la principal (master) Com ho faríem? → fes-ho

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git ((0396265...))
$ git checkout main
Previous HEAD position was 0396265 Git Ignore
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ |
```

16. Suposem que volem eliminar completament la branca la_meva_branca del git. Com ho faríem? → fes-ho. Et dona un error? Perquè? Pots utilitzar -D → fes-ho

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git branch matias -d
Deleted branch matias (was 0396265).
```

17. Ara suposem que ens demanen fer uns canvis en un altra branca «branca2» en que hem d'afegir algun text a la pàgina index.html → fes-ho.

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git checkout branca2
Switched to branch 'branca2'

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (branca2)
$ git add index.html

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (branca2)
$ git commit -m "Branca2"
[branca2 377d2f6] Branca2
1 file changed, 2 insertions(+), 1 deletion(-)

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (branca2)
$ git push
fatal: The current branch branca2 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin branca2

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (branca2)
$ ^C

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (branca2)
$ git push --set-upstream origin branca2
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 287 bytes | 287.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'branca2' on GitHub by visiting:
remote:   https://github.com/mrodriguezzsx/activitat_git/pull/new/branca2
remote:
To https://github.com/mrodriguezzsx/activitat_git.git
 * [new branch]      branca2 -> branca2
branch 'branca2' set up to track 'origin/branca2'.
```

18. Si ara volem fusionar els canvis de la branca2 a master com ho farem? → fes-ho
Un cop s'ha pujat, al github ens sortira la opcio de fer Pull Request.

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (branca2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 622 bytes | 28.00 KiB/s, done.
From https://github.com/mrodriguezzsx/activitat_git
   323051c..1433edb  main      -> origin/main
Updating 323051c..1433edb
Fast-forward
 index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Github

Crea't un compte gratuït a github per continuar la pràctica.




19. Dins de github crea un projecte «activitat_git» nou i puja el projecte local. (pista: git remote add... / git push ...) → fes-ho.

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git remote add origin https://github.com/mrodriguezzsx/activitat_git.git
```

20. Afegeix un fitxer al projecte i fes commit. Ara puja els canvis a github (git push origin...)

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 232 bytes | 116.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mrodriguezzsx/activitat_git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

21. Suposem que vols unir-te a un projecte ja existent en github. Hauries de crear una carpeta nova i descarregar-te el projecte a la carpeta buida. Crea una carpeta buida i descarrega't el projecte activitat_git (pista: clone)

← → ▾ ↑  > Este equipo > Escritorio > JaumeBalmes > M8 > UF4 > carpeta buida					▼	🔄
★ Acceso rápido		Nombre	Fecha de modificación	Tipo	Tamaño	
 Escritorio		 activitat_git	18/05/2023 4:03	Carpeta de archivos		

22. Suposem que ara un altre company fa canvis en el projecte (per a simular-ho pots fer canvis des de l'altra carpeta on està el projecte com si es tractés d'un altre ordinador. Fes canvis i publica'ls a github). Com obtindries els canvis amb les diferències del github a la carpeta local? (pista pull) → fes-ho

```
mrodr@DESKTOP-9HKC1D5 MINGW64 ~/Desktop/JaumeBalmes/M8/UF4/activitat_git (main)
$ git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 622 bytes | 28.00 KiB/s, done.
From https://github.com/mrodriguezzsx/activitat_git
 323051c..1433edb main      -> origin/main
Updating 323051c..1433edb
Fast-forward
 index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```