

---

# Comandes terminal Windows

Cristina Roigé Gaja

## Table of Contents

Que és git .....	1
Instal·lar GIT a Windows .....	1
Configurar git .....	1
Crear repositori .....	3
Clonar repositoris .....	5
Espai de treball a GIT .....	6
Comandes status add dif .....	8
fixer gitignore .....	9
Comanda commit i versions .....	10
Comandes reset i checkout .....	12
Desfer un commit .....	13
Tornar a un commit anterior .....	15
Buscar un commit .....	17

## Que és git

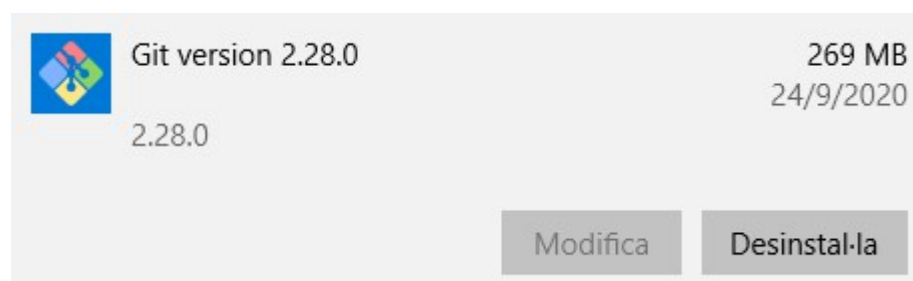
Git és un programa de control de versions que ha estat pensat per tindre una gran eficiència i confiabilitat del manteniment de versions de les nostres aplicacions.

Git l'utilitzarem per tindre control i registrar els canvis realitzats sobre un o diversos arxius de manera que si ens equivoquem o volem recuperar alguna versió antiga ho tinguem més fàcil que gestionant còpies de seguretat a pel.

## Instal·lar GIT a Windows

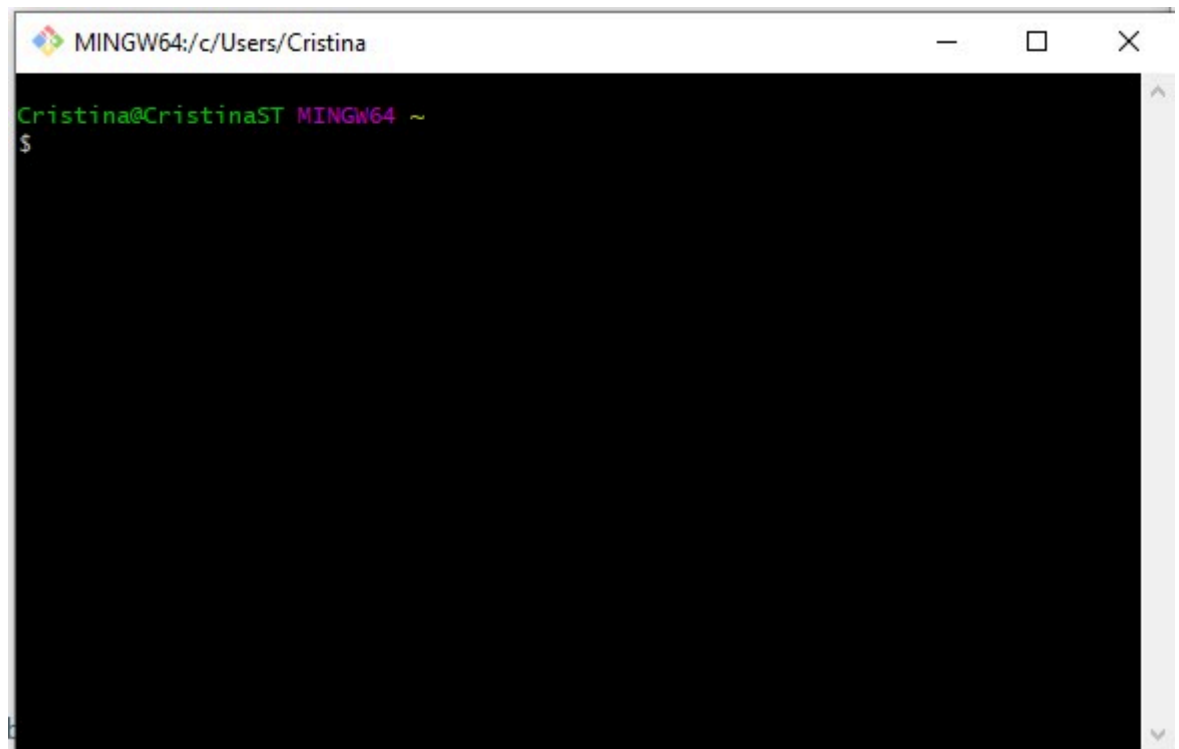
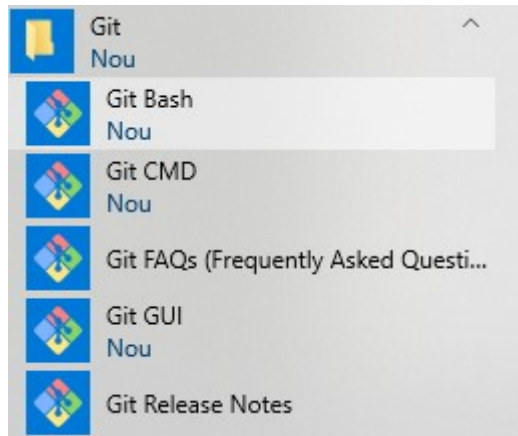
Per instal·lar GIT a windows una de les maneres que hi ha és anant a la següent pàgina <https://git-scm.com/> i descarregar la versió de Windows. un cop descarregat a baixades executarem el fitxer i el configurarem com volguem.

Un cop instal·lat podem veure la versió que tinguem instal·lada

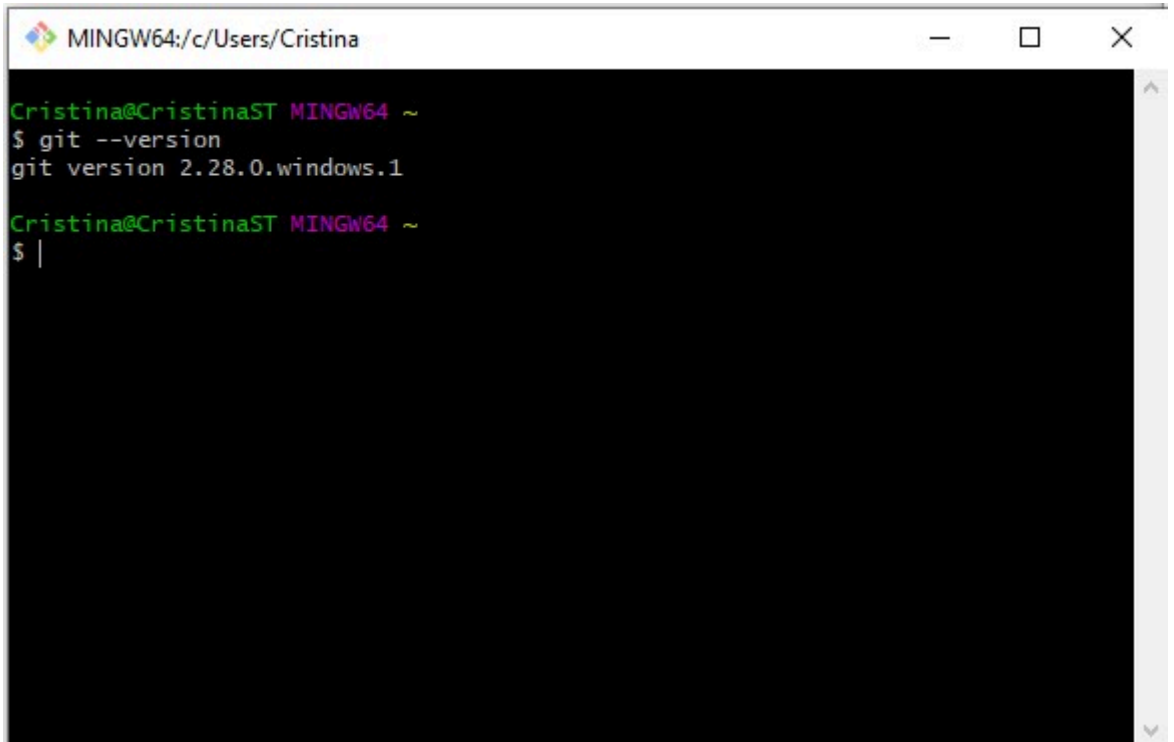


## Configurar git

Quan instal·lem GIT entre altres també s'ens instala GIT bash que es una consola.



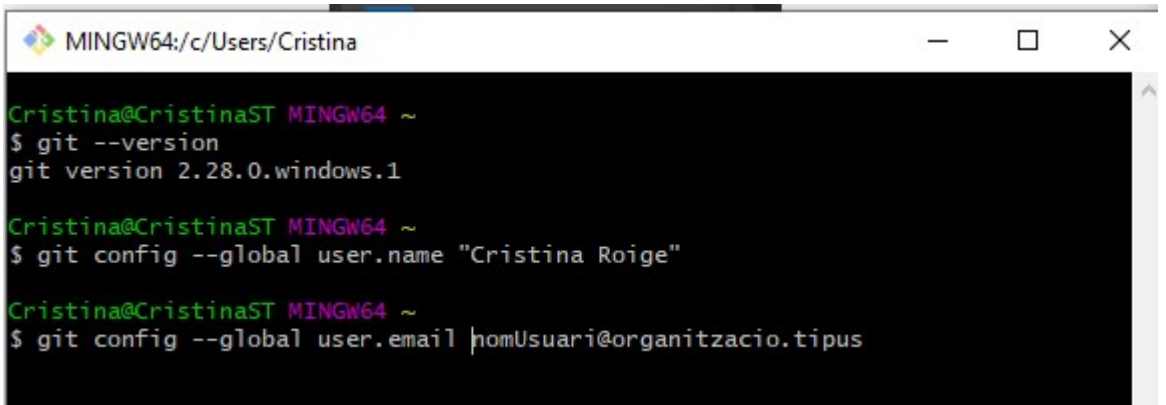
Aquí també podem veure la versió que tenim de GIT amb la comanda `GIT --version`

A terminal window titled 'MINGW64:/c/Users/Cristina' with standard Windows window controls. The prompt is 'Cristina@CristinaST MINGW64 ~'. The user enters '\$ git --version' and the output is 'git version 2.28.0.windows.1'. The prompt returns to '\$ |'.

Configurarem GIT amb el nom d'usuari i el compte de correu electrònic ja que tots els commits de GIT utilitzaran aquesta informació.

```
git config --global user.name "nom usuari"
```

```
git config --global user.email nomUsuari@organitzacio.tipus
```

A terminal window titled 'MINGW64:/c/Users/Cristina' with standard Windows window controls. The prompt is 'Cristina@CristinaST MINGW64 ~'. The user enters '\$ git --version' and the output is 'git version 2.28.0.windows.1'. The prompt returns to 'Cristina@CristinaST MINGW64 ~'. The user enters '\$ git config --global user.name "Cristina Roige"'. The prompt returns to 'Cristina@CristinaST MINGW64 ~'. The user enters '\$ git config --global user.email nomUsuari@organitzacio.tipus'.

Amb la variable `--global` GIT utilitza informació per tot el sistema. també podem configurar:

- Editor de text → `git config --global core.editor nomEditor`
- Eina per resoldre conflictes per resoldre **merge** → `git config --global merge.tool nomEina`

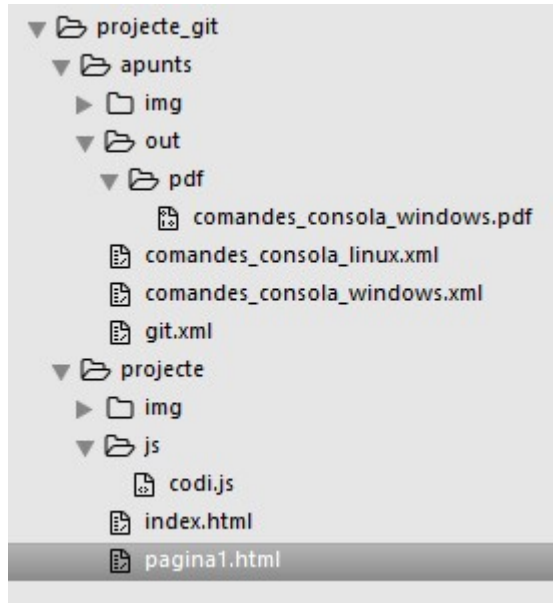
Per comprovar la configuració hem de fer la següent comanda → `git config --list`

Per poder veure les comandes que té GIT → `git help`

## Crear repositori

Per crear un repositori el primer que hem fet és crear una carpeta a qualsevol lloc que volguem i dins aquesta hem fet els següents fitxers:

- Carpeta apunts en aquesta carpeta hi hauran els apunts que estic fent per GIT.
- Carpeta projecte on hi haurà un exemple de projecte web



De moment tenim text a index.html i pagina1.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"/>
  <title>Hola m&oacute;n</title>
  <script type="text/javascript" src="js/codi.js"></script>
</head>
<body>
  <div><h1>Index git</h1></div>
  <div>
    <ul>
      <li><a href="pagina1.html">Pàgina 1</a></li>
    </ul>
  </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"/>
  <title>Hola m&ocute;n</title>
  <script type="text/javascript" src="js/codi.js"></script>
</head>
<body>
  <div><h1>Pagina 1</h1></div>
  <div>
    <ul>
      <li><a href="index.html">inici</a></li>
    </ul>
  </div>
</body>
</html>
```

Per iniciar un repositori (projecte) anirem a la carpeta que tinguem el projecte:

```
Cristina@CristinaST MINGW64 ~
$ cd Desktop/projecte_git/
```

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git
$ ls
apunts/  projecte/
```

Per iniciar un repositori hem d'escriure la comanda de GIT → `git init` podem veure que s'ens ha creat una carpeta oculta que es diu `.git`.

Per llistar els fitxer la comanda és `ls` i si volem veure els fitxers ocults `ls -la`

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git
$ git init
Initialized empty Git repository in C:/Users/Cristina/Desktop/projecte_git/.git/

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ ls
apunts/  projecte/
```

```
.git
apunts
projecte
```

## Clonar repositoris

Podem crear un repositori clonant un que ja existeixi i es pot fer de dues formes o clonant un repositori local o bé un repositori remot.

Clonar repositori local → `git clone ./rutaRelativa nomNovaCarpeta`. es clonarà però es clonarà buit.

```
Cristina@CristinaST MINGW64 ~/Desktop
$ git clone ./projecte_git/ projecte_git_clonat
Cloning into 'projecte_git_clonat'...
warning: You appear to have cloned an empty repository.
done.

Cristina@CristinaST MINGW64 ~/Desktop
$ cd projecte_git_clonat/

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git_clonat (master)
$ ls

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git_clonat (master)
$ ls -la
total 20
drwxr-xr-x 1 Cristina 197609 0 set. 24 19:01 ./
drwxr-xr-x 1 Cristina 197609 0 set. 24 19:01 ../
drwxr-xr-x 1 Cristina 197609 0 set. 24 19:01 .git/
```

Per clonar un repositori que tinguem a internet:

`git clone https://github.com/nomrepositori/nomProjecte.git ./nomNovaCarpeta`.

Podem veure que el nostre repositori es clonarà.

```
MINGW64:/c/Users/Cristina/Desktop/projecte_git_clonat/projecte_remot_clonat

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git_clonat
$ git clone https://github.com/cristinaRoige/apuntsGit.git ./projecte_remot_clonat
Cloning into './projecte_remot_clonat'...
remote: Enumerating objects: 79, done.
remote: Total 79 (delta 0), reused 0 (delta 0), pack-reused 79
Unpacking objects: 100% (79/79), 12.82 KiB | 2.00 KiB/s, done.

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git_clonat
$ ls -la
total 20
drwxr-xr-x 1 Cristina 197609 0 set. 24 19:25 ./
drwxr-xr-x 1 Cristina 197609 0 set. 24 19:25 ../
drwxr-xr-x 1 Cristina 197609 0 set. 24 19:26 projecte_remot_clonat/

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git_clonat
$ cd projecte_remot_clonat/

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git_clonat/projecte_remot_clonat (master)
$ ls
arxiuProva.txt index.html nouArxiu.txt p1.html styles.css t1.txt

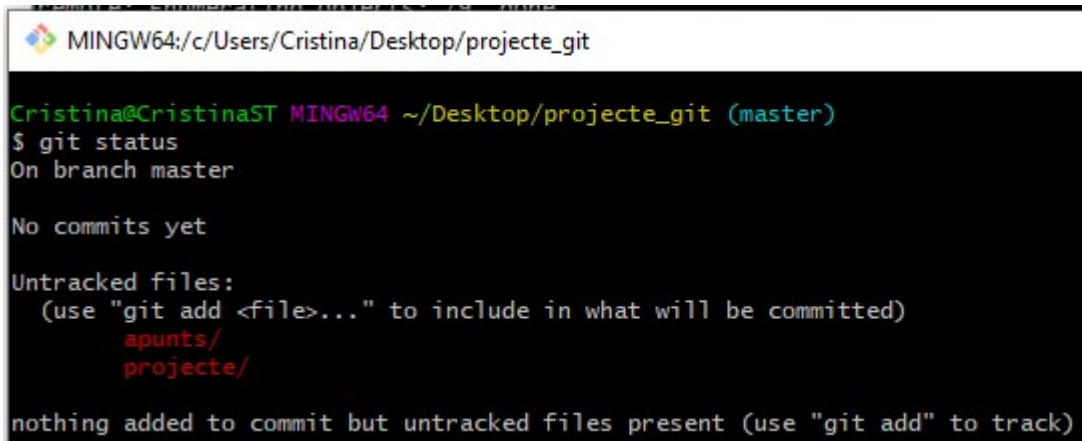
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git_clonat/projecte_remot_clonat (master)
$ |
```

## Espai de treball a GIT

La comanda `git status` mostra tot el que hi ha dins de l'index.

Ens indica a quina branca estem, en aquest cas la branch master ens diu que no hem fet cap commit i els untracked files (fitxers que no estan inclosos per fer el commit, directori de treball).





```

MINGW64:/c/Users/Cristina/Desktop/projecte_git

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git status
On branch master

No commits yet

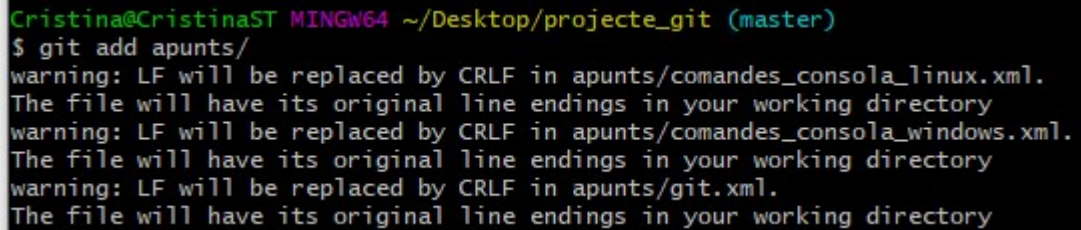
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        apunts/
        projecte/

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

```

Si volem fer un commit amb algun d'aquest fitxers el que hem de fer es fer servir la següent comanda:

```
git add nomCarpetaODirectori
```

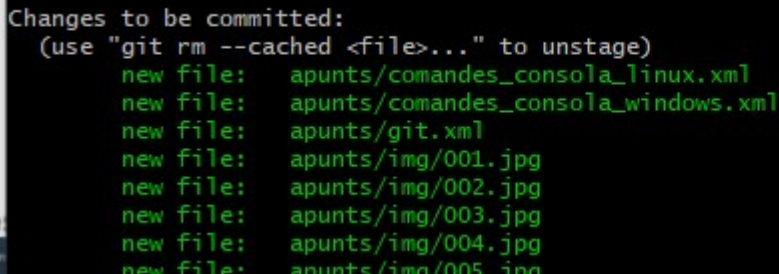


```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git add apunts/
warning: LF will be replaced by CRLF in apunts/comandes_consola_linux.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in apunts/comandes_consola_windows.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in apunts/git.xml.
The file will have its original line endings in your working directory

```

Si ara fem un `git status` podem veure la safata de sortida (staging index) els fitxers i carpetes que estan apunt de rebre el commit.

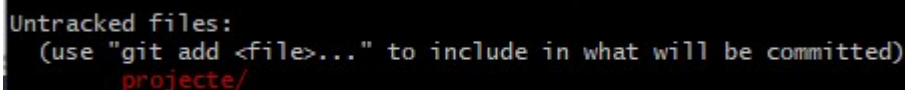


```

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   apunts/comandes_consola_linux.xml
        new file:   apunts/comandes_consola_windows.xml
        new file:   apunts/git.xml
        new file:   apunts/img/001.jpg
        new file:   apunts/img/002.jpg
        new file:   apunts/img/003.jpg
        new file:   apunts/img/004.jpg
        new file:   apunts/img/005.jpg

```

També podem veure untracked files. aquestes carpetes i fitxers estan pendents de passar a la safata de sortida i si ara mateix fessim un commit aquest fitxers no s'inclourien en aquest.



```

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

```

Si volguessim esborrar algun fitxer de la safata de sortida i no incloure'l en el següent commit hauriem de fer servir la comanda:

```
git rm --cached <nomArxiuODirectori>
```

Si volguessim afegir tots els fitxers de cop al següent commit fariem servir la comanda

```
git add .
```

## Comandes status add dif

Si fem dos canvis als fitxers index.html i pagina1.html i després fem un GIT status podem veure que aquests dos fitxers estan al nostre directori de treball i no estaran inclosos en el commit. si fem GIT add al fitxer index.html podem veure que només ens queda el fitxer pagina1.html al directori de treball.

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   projecte/index.html
        modified:   projecte/pagina1.html
```

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   projecte/index.html
        modified:   projecte/pagina1.html
```

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git add projecte/index.html
```

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   projecte/pagina1.html
```

Si no ens agraden els canvis que hem fet en aquesta pàgina podem utilitzar la comanda

```
git checkout pagina1.html
```

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git checkout projecte/pagina1.html
Updated 1 path from the index
```

podem veure que a la safata de sortida ja no ens hi queda res.

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$
```

Si tornem a fer canvis a la pagina1.html guardem i a la consola tornem a fer GIT status podem veure que la pagina1.html s'ha modificat.

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   projecte/pagina1.html

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
```

Si volem veure els canvis que s'han produït hem d'utilitzar la comanda

```
git diff nomCarpetaOArxiu
```



```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git diff projecte/pagina1.html
diff --git a/projecte/pagina1.html b/projecte/pagina1.html
index 1b6081a..884e682 100644
--- a/projecte/pagina1.html
+++ b/projecte/pagina1.html
@@ -21,6 +21,6 @@
         <li><a href="index.html">inici</a></li>
     </ul>
 </div>
-    <div class="clear"><p>Canvi 1</p></div>
+    <div class="clear"><p>Canvi a la pàgina 1.</p></div>
</body>
</html>
\ No newline at end of file

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)

```

En vermell podem veure com estava abans i en verd com està ara.

## fitxer gitignore

El fitxer gitignore serveix perquè una carpeta o arxiu no es guardi mai en un commit.

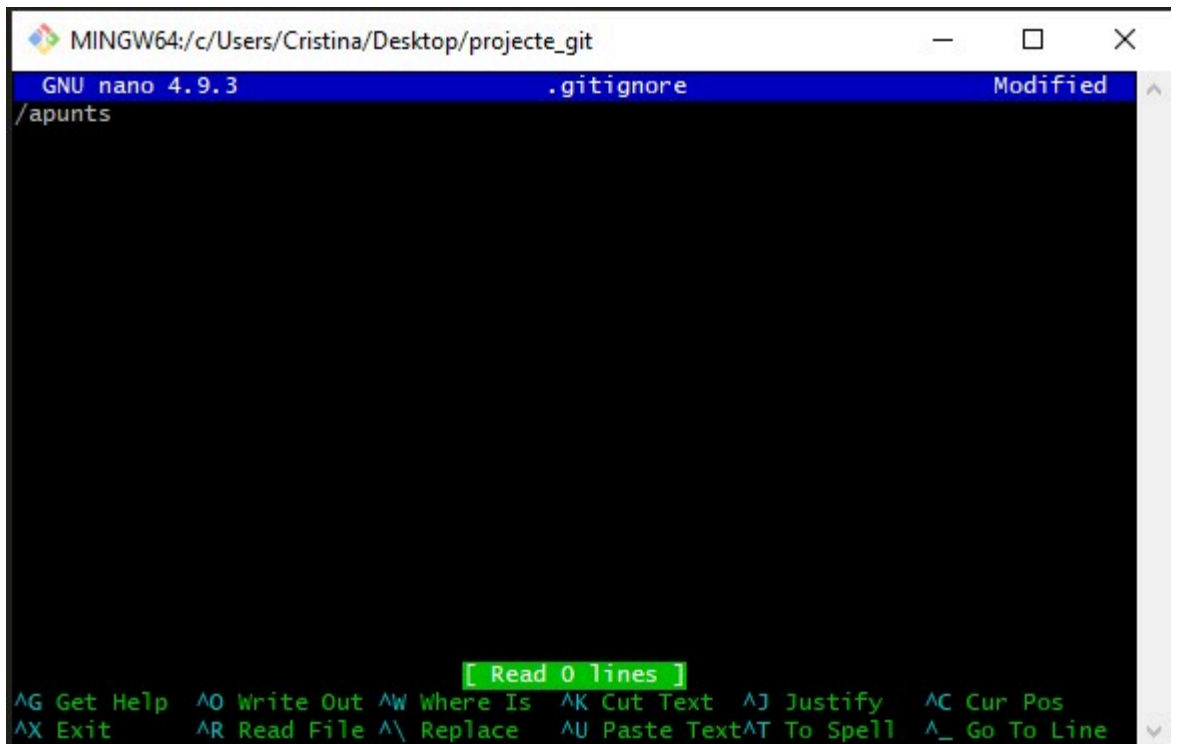
```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ touch .gitignore

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ nano .gitignore

```

hem d'escriure la ruta de la carpeta des del directori de treball



Si fem `git status` ara podem veure aquest fitxer que podem fer commit i que evita que els fitxers que estan dins no s'afegeixin mai.

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

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
```

## Comanda commit i versions

Un commit és una versió del programa actual, un punt de partida al nostre codi que podrem tornar sempre que volguem i comparar-lo amb altres versions que tinguem.

Un cop tinguem tots els fitxers a la safata de sortida després de fer un add hem d'utilitzar la següent comanda:

```
git commit -m "comentari identificatiu del comit actual"
```

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git commit -m "commit inicial"
```

Si fem git status veiem que no hi ha cap commit per fer.

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Si fem servir la comanda

git log --oneline veurem aquest commit.

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
708b461 (HEAD -> master) commit inicial
```

Si fem git log podrem veure tota la informació del commit

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log
commit 708b4614c7d0fa92bf82ddd0be1c769fb0b1d039 (HEAD -> master)
Author: Cristina Roige <crisinaroige75@gmail.com>
Date: Thu Sep 24 22:23:11 2020 +0200

    commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
```

Si fem canvis al fitxer index.html i al fitxer paginal.html podem *si estem molt segurs que podem fer els canvis* podem fer servir la comanda git commit -am "comentari identificatiu del commit actual.

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   projecte/index.html
        modified:   projecte/pagina1.html

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

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git commit -am "afegim paragraf index.html i pagina1.html"
[master 40dfa13] afegim paragraf index.html i pagina1.html
2 files changed, 2 insertions(+), 1 deletion(-)

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)

```

Si tornem a fer `git log` podem veure l'història dels commits.

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log
commit 40dfa1304c762bc58b104a04cb87faf5208e9bd7 (HEAD -> master)
Author: Cristina Roige <crisinaroige75@gmail.com>
Date: Thu Sep 24 22:35:34 2020 +0200

    afegim paragraf index.html i pagina1.html

commit 708b4614c7d0fa92bf82ddd0be1c769fb0b1d039
Author: Cristina Roige <crisinaroige75@gmail.com>
Date: Thu Sep 24 22:23:11 2020 +0200

    commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)

```

Per comparar dos commits es fa servir la comanda `git diff idCommit1 idCommit2`

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
40dfa13 (HEAD -> master) afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git diff 40dfa13 708b461
diff --git a/projecte/index.html b/projecte/index.html
index 1b55...

```

Podem veure els canvis que hi ha hagut entre els dos commits. en vermell el que s'ha canviat en verd el que s'ha eliminat.

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git diff 40dfa13 708b461
diff --git a/projecte/index.html b/projecte/index.html
index 824af0f..65ea575 100644
--- a/projecte/index.html
+++ b/projecte/index.html
@@ -22,6 +22,6 @@
         </ul>
     </div>
     <div class="clear"><p>Canvi 1</p></div>
-    <div class="clear"><p>Hem afegit un paràgraf</p></div>
+
</body>
</html>
\ No newline at end of file
diff --git a/projecte/pagina1.html b/projecte/pagina1.html
index 8d3819b..884e682 100644
--- a/projecte/pagina1.html
+++ b/projecte/pagina1.html
@@ -22,6 +22,5 @@
     </ul>
</div>
<div class="clear"><p>Canvi a la pàgina 1.</p></div>
-    <div class="clear"><p>Hem afegit un paràgraf</p></div>
</body>
...skipping...
diff --git a/projecte/index.html b/projecte/index.html
index 824af0f..65ea575 100644
--- a/projecte/index.html
+++ b/projecte/index.html
@@ -22,6 +22,6 @@
         </ul>
     </div>
     <div class="clear"><p>Canvi 1</p></div>
-    <div class="clear"><p>Hem afegit un paràgraf</p></div>
+
</body>
</html>
\ No newline at end of file
diff --git a/projecte/pagina1.html b/projecte/pagina1.html
index 8d3819b..884e682 100644
--- a/projecte/pagina1.html
+++ b/projecte/pagina1.html
@@ -22,6 +22,5 @@
     </ul>
</div>
<div class="clear"><p>Canvi a la pàgina 1.</p></div>
-    <div class="clear"><p>Hem afegit un paràgraf</p></div>
</body>
</html>
\ No newline at end of file
~
~

```

## Comandes reset i checkout

Si fem un canvi a un fitxer, per exemple a index.html si per error fem la comanda `git add`. I ens adonem que ens hem equivocat I el volem tornar al directori de treball podem fer servir la comanda `git reset HEAD nomFitxer` o bé també podem fer servir la comanda `git reset` . o bé `git checkout nomFitxer`



```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   projecte/index.html

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

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git add .

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   projecte/index.html

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$

```

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git reset HEAD projecte/index.html
Unstaged changes after reset:
M       projecte/index.html

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   projecte/index.html

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

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$

```

## Desfer un commit

Per veure la comanda reset que es fa servir per desfer un commit hem fet diverses modificacions i diversos commits

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
cc6230e (HEAD -> master) canvi per veure els resets 3
713a3dd canvi per veure els resets 3
9697feb canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

```

La comanda `git reset --hard HEAD` reseteja el directori de treball a l'últim estat al que estavem treballant.

Si volem desfer el últim commit, en el nostre cas cc6230e hem d'escriure la comanda.

```
git reset --hard HEAD~numCommitsQueVolemTornarEnrera
```

Per exemple en el nostre cas `$ git reset --hard HEAD~1` esborrarà l'últim commit.

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
cc6230e (HEAD -> master) canvi per veure els resets 3
713a3dd canvi per veure els resets 3
9697feb canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git reset --hard HEAD~1
HEAD is now at 713a3dd canvi per veure els resets 3

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
713a3dd (HEAD -> master) canvi per veure els resets 3
9697feb canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial
```

Hem de tindre en compte que els commits es compten a partir de 0.

Si volem esborrar uns commit farem `git reset --hard idComit` això ens esborrarà des del HEAD fins al commit anterior al que hem posat de ID

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
713a3dd (HEAD -> master) canvi per veure els resets 3
9697feb canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git reset --hard c7f05b5
HEAD is now at c7f05b5 canvi per veure els resets 1

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
c7f05b5 (HEAD -> master) canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
```

Si no fem servir el parametre hard la diferència és que desfem el commit però el codi queda intacte als fitxers.



```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
b07ba7b (HEAD -> master) canvi per veure els resets 4
31f2096 canvi per veure els resets 3
e4cbdd7 canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git diff b07ba7b 31f2096
diff --git a/projecte/index.html b/projecte/index.html
index ed740f8..db6adca 100644
--- a/projecte/index.html
+++ b/projecte/index.html
@@ -27,6 +27,5 @@
     <div class="clear"><p>Hem afegit el paràgraf 3</p></div>
     <div class="clear"><p>Hem afegit el paràgraf 4</p></div>
     <div class="clear"><p>Hem afegit el paràgraf 5</p></div>
-    <div class="clear"><p>Hem afegit el paràgraf 6</p></div>
   </body>
 </html>
\ No newline at end of file

```

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git reset 31f2096
Unstaged changes after reset:
M   projecte/index.html

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
31f2096 (HEAD -> master) canvi per veure els resets 3
e4cbdd7 canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$

```

## Tornar a un commit anterior

Si volem canviar el HEAD dels commits I que aquest sigui el nostre últim commit hem de fer servir la comanda

```
git checkout idCommit
```

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git log --oneline
a2d2bfd (HEAD -> master) recuperar commit 3
7659c74 recuperar commit 2
a470d65 recuperar commit 1
53fd94b git.xml tornar commit anterior
31f2096 canvi per veure els resets 3
e4cbdd7 canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git (master)
$ git checkout 31f2096
Note: switching to '31f2096'.

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 31f2096 canvi per veure els resets 3
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git ((31f2096...))

```

Els commits anteriors no han desaparegut i els podem tornar a recuperar.

```

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git ((31f2096...))
$ git log --oneline
31f2096 (HEAD) canvi per veure els resets 3
e4cbdd7 canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git ((31f2096...))

```

```
Cristina@CristinaST MINGW64 ~/Desktop/projecte_git ((31f2096...))
$ git checkout a2d2bfd
Previous HEAD position was 31f2096 canvi per veure els resets 3
HEAD is now at a2d2bfd recuperar commit 3

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git ((a2d2bfd...))
$ git log --oneline
a2d2bfd (HEAD, master) recuperar commit 3
7659c74 recuperar commit 2
a470d65 recuperar commit 1
53fd94b git.xml tornar commit anterior
31f2096 canvi per veure els resets 3
e4cbdd7 canvi per veure els resets 2
c7f05b5 canvi per veure els resets 1
067bf36 git.xml section 10
10d1bd4 actualitzacio fitxer git.xml
40dfa13 afegim paragraf index.html i pagina1.html
708b461 commit inicial

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git ((a2d2bfd...))
$ git status
HEAD detached at a2d2bfd
nothing to commit, working tree clean

Cristina@CristinaST MINGW64 ~/Desktop/projecte_git ((a2d2bfd...))
```

## Buscar un commit