

Pràctica 2) Fonaments del funcionament del sistema de control de versions (SCV) Git. Ordres bàsiques del sistema de control de versions Git.

Documentació

- 1- Conceptes bàsics de Git: Snapshots. Integritat. Operacions locals. Les 3 àrees de residència dels fitxers.
<http://git-scm.com/book/en/v2> → 1.3 What is Git?
<http://git-scm.com/book/es/v2> → 1.3 Fundamentos de Git
- 2- El cicle de vida i estat dels fitxers. Enregistrant canvis en el dipòsit.
<http://git-scm.com/book/en/v2> → 2.2 Recording changes to the Repository
<http://git-scm.com/book/es/v2> → 2.2 Guardando cambios en el Repositorio
- 3- Configuració inicial de Git
<http://git-scm.com/book/en/v2> → 1.6 First-Time Git Setup
<http://git-scm.com/book/es/v2> → 1.6 Configurando Git por primera vez
- 4- Ordres bàsiques de Git
<http://git-scm.com/book/en/v2> → 2 Git Basics (2.1 a 2.5 per aquesta pràctica)
<http://git-scm.com/book/es/v2> → 2 Fundamentos de Git (2.1 a 2.5 per aquesta pràctica)
- 5- Altres enllaços sobre ordres bàsiques de Git.
 - a) Tutorial pràctic en línia de Github i ordres Git: <https://try.github.io/levels/1/challenges/1>
 - b) Explicació resumida d'algunes de les ordres més importants: <http://rogerdudler.github.io/git-guide/>
 - c) **HEAD vs master/main vs origin**: <https://stackoverflow.com/questions/8196544/what-are-the-git-concepts-of-head-master-origin> (Veure la primera resposta d'en Matt Greer)
 - d) Guia visual: <http://marklodato.github.io/visual-git-guide/index-en.html>
 - e) Què és **origin**? → <http://stackoverflow.com/questions/9529497/what-is-origin-in-git> (Primera resposta editada per en Peter Mortensen)
 - f) Canviar el nom de branca **master** a **main** [aquí](#)

1- Conceptes bàsics - Documentació (1)

- a) Què és un snapshot o instantània?

A CADA INSTANTANIA (SNAPSHOT) LO QUE FEM ÉS PUJAR AL DIPÒSIT ELS ARXIUS MODIFICATS/CREATS, EN CAS DE QUE ES FACI UNA MODIFICACIÓ A UN ARXIU, ES PUJA AQUEST NOU FITXER, PERÒ ELS FITXERS NO MODIFICATS ES TORNEN EN ENLLAÇOS, CADA SNAPSHOT ES FA AL REALITZAR UN COMMIT.

- b) Quan GIT realitza una instantània, que fa amb els fitxers que no han estat modificats?

QUAN ES FA UN COMMIT I HI HA ALGUN ARXIU NO HA ESTAT MODIFICAT, GIT CREA UN ENLLAÇ A L'ARXIU, DES DE L'ANTIC COMMIT FINS AL NOU.

- c) Què és i quina és la utilitat de la suma de comprovació o checksum?. Quin és el seu format?.

EL CHECKSUM ÉS UN CODI HASH EN FORMAT HEXADECIMAL DE 40 BYTES MITJANÇANT L'ALGORISME SHA-1 O POSTERIOR, EL QUAL ENS IDENTIFICA UNA SNAPSHOT, AL SER UN CODI HASH ENS ASSEGURA LA INTEGRITAT DE LA INSTANTANIA, AIXÍ PODENT DETECTAR QUALESVOL MODIFICACIÓ IRREGULAR.

d) Quins són els tres estats en els quals es pot trobar un arxiu d'un projecte?

HI HA TRES TIPUS D'ESTAT PELS FITXERS:

- 1: EN MODE «COMMITTED» QUE VOL DIR «CONFIRMAT».**
- 2: EN MODE «STAGED» QUE VOL DIR EN «PREPARACIÓ».**
- 3: EN MODE «MODIFIED» QUE VOL DIR «MODIFICAT».**

e) Què significa que un fitxer està en estat "Modified" (modificat)?

UN ARXIU EN MODE MODIFICAT VOL DIR QUE HA ESTAT MODIFICAT DES DE LA DARRERA VEGADA DEL COMMIT I ENCARA NO S'HA FET NI L'ADD NI EL COMMIT PER PUJAR LA NOVA VERSIÓ AL DIPÒSIT, EL NOM DE L'ARXIU APAREIXERÀ EN VERMELL.

f) Què significa que un fitxer està en estat "Staged" (preparat)?

UN ARXIU EN MODE STAGED VOL DIR QUE HA ESTAT MODIFICAT, A PART D'HAVER ESTAT AFEGIT A L'ÀREA DE PREPARACIÓ, AMB LA QUAL COSA ES PUJARÀ EN EL SEGÜENT COMMIT. EL FITXER SEGUIRÀ Estant A L'ESTAT DE MODIFIED, PERÒ ESTARÀ ESCRIT EN COLOR VERD.

g) Què significa que un fitxer està en estat "Committed" (consignat o confirmat)?

UN ARXIU EN MODE COMMITED VOL DIR QUE ELS CANVIS REALITZATS HAN ESTAT DESATS DEFINITIVAMENT EN EL DIPÒSIT LOCAL DE GIT, EL QUAL ES DIU «.git».

h) Quines són les 3 seccions o àrees d'un projecte Git?

- 1: WORKING DIRECTORY «DIRECTORI DE TREBALL».**
- 2: STAGING AREA «ÀREA DE PREPARACIÓ», ESTÀ AL .git.**
- 3: REPOSITORY «DIPÒSIT GIT», ESTÀ AL .git.**

i) Què és i que hi ha "Working directory" (o directori de treball)? Quin és el seu propòsit?

EL WORKING DIRECTORY ÉS EL DIRECTORI ON ES TROBEN ELS ARXIS I CARPETES DEL PROJECTE EN EN QUE S'ESTÀ TREBALLANT ACTUALMENT, NORMALMENT ÉS UNA CÒPIA DEL DARRER COMMIT. AQUÍ ÉS ON ESTÀ LOCALITZAT EL DIRECTORI .git.

j) Què és la "Staging area" (o àrea de preparació)?

LA STAGING AREA ÉS UNA CARPETA LOCALITZADA AL .git (objects), A PART DE SER L'ARXIU .git/index EL CONTINGUT D'AQUEST FITXER I DIRECTORI NO ES POT LLEGIR BÉ, A CAUSA QUE ÉS UN BINARI. EN LA QUAL ÉS ON ES DESEN ELS ARXIS QUE ES PUJARAN EN EL REPOSITORI EN EL SEGÜENT COMMIT.

k) Què hi ha al "Git directory" (o directori Git). A on es troba?

EL DIRECTORI .git ESTÀ LOCALITZAT DINS DEL DIRECTORI DE TREBALL DEL PROJECTE, I ES CREA CADA COP QUE ES FA UN git init.

AQUÍ DINS ES POT TROBAR:

- 1: ELS ARXIS PUJATS PER TOTS ELS COMMITS REALITZATS.**
- 2: ARXIS I DIRECTORIS PUJATS AL STAGING AREA.**
- 3: METADADES DE TOT TIPUS, COM ARA EL CODI HASH D'UN COMMIT.**

l) Què hem de fer afegir un fitxer modificat del directori de treball a l'àrea de preparació?

HEM DE FER UN «git add <NOM FITXER>».

m) Què hem de fer per passar de un fitxer de l'àrea de preparació al dipòsit Git?

HEM DE FER UN «git commit -m «<MISSATGE>»».

n) Quan realitzem una instantània (o snapshot)?

ES FA EN EL MOMENT DE FER UN COMMIT. DENÇÀ QUE A VEGADES S'UTILITZA COMMIT COM A SINÒNIM DE SNAPSHOT, REALMENT IN COMMIT ÉS L'ACCIÓ DE CREAR UNA SNAPSHOT, MENTRES QUE UNA SNAPSHOT ÉS EL CONJUNT D'ARXIS I CARPETES D'UN PROJECTE PUJATS AL REPOSITORI A L'HAVER FER EL COMMIT.

2- Cicle de vida i estat dels fitxers. Enregistrant canvis en el dipòsit - Documentació (2)

a) Què és un fitxer "untracked" (no seguit o no controlat)?

VOL DIR QUE PER GIT NO HA TREBALLAT MAI AMB ELLS (NI TÉ CONSTÀNCIA DE LA SEVA EXISTÈNCIA), JA QUE NO S'HA FET CAP «git add» NI «git commit» AMB LA QUAL COSA, S'HA DE D'AFEGIR O FER UN COMMIT. QUALESEVOL ARXIU QUE ESTIGUI UNTRACKED, NO ES PUJARÀ EN FER EL PRÒXIM COMMIT.

b) Què és un fitxer "tracked" (seguit o controlat)?

SI EN L'ANTERIOR CAS VOLIA DIR QUE NO ESTAVA AFEGIT A GIT, ARA VOL DIR QUE S'HA FET UN «git add» O «git commit» AMB LA QUAL COSA, GIT SAP DE LA SEVA EXISTÈNCIA.

c) En quins estats es pot trobar un fitxer "tracked"?

UN ARXIU ES POT TROBAR EN ELS SEÜENTS TRES ESTATS:

- 1.) MODIFIED (MODIFICAT) → SI L'HEM CAMBIAT PERÒ NO HE FET NI UN ADD NO COMMIT.**
- 2. NO MODIFIED (NO MODIFICAT) → UN ARXIU QUE NO ESTAVA EN PREPARACIÓ I L'HE PUJAT AL DIRECTORI GIT, PERÒ AL NO SER CAMBIAT, ES QUEDA IGUAL EN EL REPOSITORI I EN EL DIRECTORI DE TREBALL.**
- 3. STAGGED (EN PREPARACIÓ) → ARXIU MODIFICAT QUE S'HA FET UN ADD.**

d) Quina diferència hi ha entre un fitxer modificat i un fitxer en preparació?

UN ARXIU DE PREPARACIÓ HA ESTAT MODIFICAT, I S'HA FET UN ADD, MENTRES QUE EL MODIFICAT HA ESTAT CAMBIAT, PERÒ SENSE HAVER FER UN ADD.

e) Què hem de fer per poder "consignar" (commit) d'un fitxer "sense seguiment" (untracked)?

PER POER FER UN COMMIT, S'HA DE FER UN «git add <FILE>» ABANS.

f) Un fitxer modificat, si no està en preparació (staged) pot quedar consignat (commit)?

NO, NO ES POT, JA QUE PREVIAMENT S'HA D'HAVER FET UN «git add», DESPRÉS DE FER UN COMMIT, ELS ARXIS ESTARAN EN ESTAT UNMODIFIED, JA QUE ELS ARXIS DEL DIRECTORI ACTUAL ESTARAN IGUAL QUE A L'ÚLTIMA VERSIÓ DEL COMMIT DEL REPOSITORI.

g) Després de consignar tots els arxius d'un projecte, quin és el seu estat?

COM ANTERIORMENT JA HE DIT, RESUMINT ESTARÀ EN UNMODIFIED, JA QUE TÉ L'ÚLTIMA VERSIÓ DEL COMMIT DEL REPOSITORI

h) Quin és l'estat de tots els arxius d'un projecte quan es fa un clonatge des d'un dipòsit?

TOTS ELS ARXIUS ESTARAN UNMODIFIED, JA QUE ELS ARXIUS SERAN ELS MATEIXOS I TINDRAN LA MATEIXA VERSIÓ QUE LA DEL REPOSITORI.

3- Apuntador main, HEAD i origin/main - Documentació (5.c - 5d - 5.e)

a) A on apunta per defecte l'apuntador main (master)?. Què és la branca main(master)?

UN PUNTER SERVEIX PER IDENTIFICAR UN COMMIT DINS D'UNA BRANCA. AQUEST S'IDENTIFICA AMB UN CODI HASH, EXISTEIXEN DIVERSOS APUNTADORS ESPECIALS QUE TENEN UN NOM PROPI, COM ARA PER EXEMPLE «main».

AQUEST PUNTER «main» SEMPRE APUNTA A L'ÚLTIM COMMIT D'UNA BRANCA. UNA BRANCA ÉS UN CONJUNT DE COMMITS REALITZATS UN DARRERE L'ALTRE, EL NOM D'AQUESTA BRANCA ÉS EL MATEIX QUE EL NOM DEL PUNTER QUE APUNTA A L'ÚLTIM COMMIT.

b) A on apunta normalment HEAD?.

AQUEST ÉS UN APUNTADOR AMB UN NOM PROPI, NO IDENTIFICAT AMB CAP CODI HASH. AQUEST APUNTADOR ES POT MOURE, AMB LA QUAL COSA, S'ARRIBA A AQUESTA CONCLUSIÓ:

- **ELS PUNTERS DE CODI HASH SÓN FIXES I INAMOBIBLES.**
- **EL PUNTER «main» ES POT MOURE, PERÒ NOMÉS ES MOU PER PASSAR A APUNTAR A L'ÚLTIM COMMIT.**
- **EL PUNTER «head» ES POT APUNTAR A ON VOLGUEM.**

c) Què significa que HEAD està "detached" (deslligat o separat)?

VOL DIR QUE L'HEM CANVIAT DE POSICIÓ I QUE NO APUNTA AL MATEIX QUE EL «main», JA QUE PER DEFECTE, «head» APUNTA AL MATEIX QUE L'APUNTADOR «main». DIENT-HO AMB ALTRES PARAULES, QUE PER DEFECTE «head» ESTÀ LLIGAT A L'APUNTADOR «main».

d) Què és origin? A on apunta origin/main (origin/master)? A on apunta origin/HEAD?

ORIGIN ÉS UNA ETIQUETA QUE IDENTIFICA L'URL DEL DIPÒSIT REMOT, O SIGUI QUE ÉS PER NO ESCRIURE «https://github.com/...».

L'APUNTADOR «origin/main» ÉS LA BRANCA MAIN DEL DIPÒSIT main, ELS ÚLTIMS APUNTADORS ANOMENATS «head» I «main» AQUEST ÚLTIM APUNTA AL DARRER COMMIT DE LA BRANCA «main», MENTRES QUE «head» ÉS UNA APUNTADOR QUE ES POT MOURE DINS D'UNA BRANCA. I FINALMENT L'APUNTADOR «origin/head», ÉS EL MATEIX PERÒ PODENT MOURE L'APUNTADOR A ON ES VULGUI.

4- Gestionant el sistema de control de versions des de l'interpret d'ordres en local - Documentació(3) i (4)

a) Si no ho has fet a la primera pràctica, crea ara dins del teu directori personal el fitxer de configuració **.gitconfig** del gestor del sistema de control de versions **Git**.

Executa per **Debian GNU/Linux** des dels del teu directori personal:

- **git config --global user.name "xxxyyy23daw2"**
- **git config --global user.email "*****@fje.edu"**
- **git config --global core.editor nano**
- **git config --global init.defaultBranch main**

Per **Windows** executa dins del teu directori personal:

- **git config --global user.name "xxxyyy23daw2"**
- **git config --global user.email "*****@fje.edu"**
- **git config --global core.editor notepad.exe**
- **git config --global init.defaultBranch main**

Recorda que:

- **xxxyyy23daw2** és el teu compte de **Github** d'acord amb les indicacions de la pràctica 1.
- **xxx** són les 3 primeres lletres del teu nom
- **yyy** són les tres primeres lletres del teu primer cognom.
- *******@fje.edu** és el teu compte de correu de l'escola llevat que hagi decidit utilitzar un altre correu.

Comprova que s'ha creat el fitxer **.gitconfig** dins del teu **directori personal**. Comprova que els paràmetres són correctes executant:

- **git config --global -list**

```
romasarca@archlinux ~-> git config --global --list
user.name=romsar23daw2
user.email=15586186.clot@fje.edu
init.defaultbranch=main
romasarca@archlinux ~-> █
```

EN AQUEST CAS, JA HO TENIA, I SIMPLEMENT HE MOSTRAT ELS PARÀMETRES QUE TENIA JA CONFIGURATS.

Si hi ha cap error, torna a executar l'ordre sense errors i la nova configuració sobreescriurà l'anterior.

b) Crea dins del teu directori personal un nou directori de nom **mitjana**. Canvia al directori **mitjana** i descarrega el fitxers **mitjana.c** i **README** que s'utilitzaran dins de la pràctica executant:

- **wget --no-hsts http://www.collados.org/daw2/m08/uf4/mitjana/mitjana.c**
- **wget --no-hsts http://www.collados.org/daw2/m08/uf4/mitjana/README**

```
romasarca@archlinux ~-> mkdir mitjana && cd mitjana
romasarca@archlinux ~/mitjana> wget --no-hsts http://www.collados.org/daw2/m08/uf4/mitjana/mitjana.c
--2023-09-27 16:37:01-- http://www.collados.org/daw2/m08/uf4/mitjana/mitjana.c
Resolving www.collados.org (www.collados.org)... 139.162.182.159
Connecting to www.collados.org (www.collados.org)[139.162.182.159]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 452 [text/x-c]
Saving to: 'mitjana.c'

mitjana.c          100%[=====]         452  --.-KB/s   in 0s

2023-09-27 16:37:01 (38.1 MB/s) - 'mitjana.c' saved [452/452]

romasarca@archlinux ~/mitjana> wget --no-hsts http://www.collados.org/daw2/m08/uf4/mitjana/README
--2023-09-27 16:37:08-- http://www.collados.org/daw2/m08/uf4/mitjana/README
Resolving www.collados.org (www.collados.org)... 139.162.182.159
Connecting to www.collados.org (www.collados.org)[139.162.182.159]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 136
Saving to: 'README'

README            100%[=====]         136  --.-KB/s   in 0s

2023-09-27 16:37:08 (12.4 MB/s) - 'README' saved [136/136]

romasarca@archlinux ~/mitjana> ls
mitjana.c  README
romasarca@archlinux ~/mitjana>
```

EN AQUEST CAS, COM ES OPT VEURE, HE CREAT EL DIRECTORI MITJANA I HI HE ENTRAT, POSTERIORMENT HE FET ELS WGET CORRESPONENTS.

c) Crea i inicialitza un nou dipòsit local Git pels fitxers del projecte **mitjana**. Comprova:

- Que s'ha creat el directori **.git** dins del directori mitja.
- Comprova l'estat del dipòsit i quin és el nom inicial de la branca principal creada. Si el nom de la branca principal és **master**, canvia-la a **main** executant: **git checkout -b main**
- L'estat dels fitxers **mitjana.c** i **README**. Explica el motiu del seu estat.

```
romasarca@archlinux ~/mitjana> git init
Initialized empty Git repository in /home/romasarca/mitjana/.git/
romasarca@archlinux ~/mitjana (main)> la
Permissions Size User      Date Modified Name
drwxr-xr-x   - romasarca 27 Sep 17:39 .git
-rw-r--r--   452 romasarca 8 Oct 2018 mitjana.c
-rw-r--r--   136 romasarca 6 Oct 2020 README
romasarca@archlinux ~/mitjana (main)> git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
romasarca@archlinux ~/mitjana (main)>
```

PER CREAR EL REPOSITORI HE FET ÚS DEL COMANDAMENT «git init», EL QUAL CREA UN «.git». PER MOSTRAR EL CONTINGUT DEL DIRECTORI FAIG ÚS DEL «la», I PER FINALITZAR, VEIG L'ESTAT DELS ARXIUS AMB «git status», EL QUAL EN AQUEST CAS ENS DIU QUE ELS ARXIUS NO ESTAN CONTROLATS, JA QUE NO S'HA FET UN «git add».

d) Afegeix (**add**) els dos fitxers a l'àrea de preparació (staged area). Comprova ara l'estat del dipòsit. Indica quin és l'estat dels fitxers **mitjana.c** i **README**.

```
romasarca@archlinux ~/mitjana (main)> git add mitjana.c README
romasarca@archlinux ~/mitjana (main)> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README
    new file:   mitjana.c

romasarca@archlinux ~/mitjana (main)> █
```

POSTERIORMENT DE FER UN «git add», ELS ARXIUS ESTAN EN L'ÀREA DE PREPARACIÓ, TAMBÉ CONEGUT COM «staged», COM ES POT VEURE, ESTAN DE COLOR VERD.

e) Modifica el fitxer **mitjana.c**. A la línia 16, canvia **scanf(" %c",&consulta);** per **scanf("%s",&consulta);**. Comprova ara l'estat del dipòsit. Indica quin és l'estat del fitxer **mitjana.c**. D'acord amb l'estat actual, en el moment de fer una consignació (**commit**), quina versió de l'arxiu s'emmagatzemarà en el dipòsit?.

```
romasarca@archlinux ~/mitjana (main)> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README
    new file:   mitjana.c

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:   mitjana.c

romasarca@archlinux ~/mitjana (main)> █
```

L'ARXIU mitjana.c, TÉ 2 VERSIONS HI HA UN ARXIU A L'ÀREA DE PREPARACIÓ QUE TENIA A LA LÍNIA 16 «scanf(" %c",&consulta);», MENTRES QUE L'ALTRE VERSIÓ QUE ESTÀ AL DIRECTORI DE TREBALL AMB LA LÍNIA «scanf("%s",&consulta);».
DACORD AMB L'ESTAT ACTUAL ES PUJARÀ AL DIPÒSIT (.git) LA PRIMERA VERSÓ DE MITJANA.C (LA QUE CONTÉ «scanf(" %c",&consulta);»).

f) Consigna (**commit**) els fitxers que es troben a l'àrea de preparació (staged area). En el moment de consignar afegeix el comentari "*Primer commit del projecte mitjana*". Comprova que a llista de fitxers afegits al dipòsit hi ha **mitjana.c** i **README**.

```
romasarca@archlinux ~/mitjana (main)> git commit -m "Primer commit del projecte mitjana"
[main (root-commit) 9fff719] Primer commit del projecte mitjana
 2 files changed, 26 insertions(+)
 create mode 100644 README
 create mode 100644 mitjana.c
romasarca@archlinux ~/mitjana (main)> █
```

```
romasarca@archlinux ~/mitjana (main)> git show --pretty="" --name-only
README
mitjana.c
romasarca@archlinux ~/mitjana (main)> █
```

A LA PRIMERA CAPTURA (LA DE L'ESQUERRA), ES POT APRECIAR COM ESTIC FENT EL COMMIT, I A L'ALTRE DE LA DRETA MOSTRO LA LLISTA D'ARXIUS AFEGITS.

g) Comprova ara l'estat del dipòsit. Indica quin és l'estat dels fitxers **mitjana.c** i **README**. Quina és la diferència entre els estats dels 2 fitxers?. Per què?.

```
romasarca@archlinux ~/mitjana (main)> git status
On branch main
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:   mitjana.c

no changes added to commit (use "git add" and/or "git commit -a")
romasarca@archlinux ~/mitjana (main)> █
```

COM EL README NO ESTAVA MODIFICAT, EL SEU ESTAT A L'ÀREA DE TREBALL ÉS EL MATEIX QUE HI HA A L'ÚLTIM COMMIT.

MENTRES QUE **mitjana.c** COM NO HAVIA FET UN «git add» DE L'ARXIU DESPRÉS DE MODIFICAR-LO, LA VERSIÓ DE L'ÀREA DE TREBALL NO ÉS LA MATEIXA QUE LA DE L'ÚLTIM COMMIT.

h) Afegeix només **mitjana.c** a l'àrea de preparació i fes una nova consignació. En el moment de consignar afegeix el comentari "*Segon commit del projecte mitjana*". Comprova novament l'estat del dipòsit. Indica el motiu del resultat mostrat per pantalla.

```
romasarca@archlinux ~/mitjana (main)> git add mitjana.c
romasarca@archlinux ~/mitjana (main)> git commit -m "Segon commit del projecte mitjana"
[main dd58732] Segon commit del projecte mitjana
 1 file changed, 1 insertion(+), 1 deletion(-)
romasarca@archlinux ~/mitjana (main)> git status
On branch main
nothing to commit, working tree clean
romasarca@archlinux ~/mitjana (main)> █
```

EN AQUEST CAS, ARA QUAN FAIG UN «git status» REBO EL MISSATGE «nothing to commit; working tree clean», JA QUE L'ESTAT DE L'ÀREA DE TREBALL ÉS EL MATEIX QUE A L'ÚLTIM COMMIT.

i) Mostra l'historial de tots els commits (consignacions) realitzats. Identifica i marca el valor del **checksum** (o suma de comprovació) que s'ha creat per cada consigació (**commit**).

```
romasarca@archlinux ~/mitjana (main)> git log
commit dd5873294e66ba9f9404d62874ab40df2c9febbe (HEAD -> main)
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Wed Sep 27 17:54:23 2023 +0000

    Segon commit del projecte mitjana

commit 9fff719743e2748d043b39894fb25460709b8bf9
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Wed Sep 27 17:45:45 2023 +0000

    Primer commit del projecte mitjana
romasarca@archlinux ~/mitjana (main)> █
```

AQUÍ HI HA 4 APUNTADORS, DOS CODIS HASH QUE APAREIXEN AL COSTAT DE «commit», «HEAD» I «main».

j) Compila el programa **mitjana.c** amb l'ajut del **README**. Comprova l'estat del dipòsit Git. Quin és l'estat del programa **mitjana**?. Per què?

```
romasarca@archlinux ~/mitjana (main)> gcc mitjana.c -o mitjana
romasarca@archlinux ~/mitjana (main)> git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    mitjana

nothing added to commit but untracked files present (use "git add" to track)
romasarca@archlinux ~/mitjana (main)>
```

L'ESTAT DE MITJANA ESTÀ EN SEGUIMENT, JA QUE EL S'HA FET UN GIT ADD PRÈVIAMENT, I ENCARA NO S'HA FET EL «git add».

k) Troba [aquí](#) el propòsit del fitxer **.gitignore**. Crea un fitxer de nom **.gitignore** dins del directori **mitjana** amb el següent contingut:

```
.gitignore
mitjana
```

Comprova ara l'estat del dipòsit. Explica perquè no hi ha res per afegir i consignar.

```
romasarca@archlinux ~/mitjana (main)> vim .gitignore
romasarca@archlinux ~/mitjana (main)> cat .gitignore
.gitignore
mitjana
romasarca@archlinux ~/mitjana (main)> git status
On branch main
nothing to commit, working tree clean
romasarca@archlinux ~/mitjana (main)>
```

AIXÒ ÉS A CAUSA DE QUE ESTÀ DINS DE «.gitignore», AIXÒ ÉS UN ARXIU EN EL QUAL S'INDICA ELS ARXIS QUE NO ES VOLEN AFEGIR EN EL MOMENT DE FER EL «git add».

l) Intenta afegir el fitxer **mitjana** a l'àrea de preparació. Ho pots fer?. Per què?

```
romasarca@archlinux ~/mitjana (main)> git add mitjana
The following paths are ignored by one of your .gitignore files:
mitjana
hint: Use -f if you really want to add them.
hint: Turn this message off by running
hint: "git config advice.addIgnoredFile false"
romasarca@archlinux ~/mitjana (main) [1]>
```

DONCS ÉS A CAUSA D'ESTAR DINS DEL «.gitignore», O SI ÉS VOLGUÉS AFEGIR, S'HAURIA DE FER SERVIR L'OPCIÓ «-f» PER FER-HO A LA FORÇA.

m) Modifica el fitxer **mitjana.c**. A la línia 4, canvia **floatat suma=0;** per **floatat suma=0.0;**. Comprova ara l'estat del dipòsit i del fitxer **mitjana.c**. Afegeix el fitxer **mitjana.c** a l'àrea de preparació. Comprova ara l'estat del dipòsit i del fitxer **mitjana.c**.

```
romasarca@archlinux ~/mitjana (main)> vim mitjana.c
romasarca@archlinux ~/mitjana (main)> grep "float suma" mitjana.c
float suma=0.0;
romasarca@archlinux ~/mitjana (main)> git status
On branch main
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:   mitjana.c

no changes added to commit (use "git add" and/or "git commit -a")
romasarca@archlinux ~/mitjana (main)> git add mitjana.c
romasarca@archlinux ~/mitjana (main)> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   mitjana.c

romasarca@archlinux ~/mitjana (main)> █
```

COM ES POT APRECIAR, HE MODIFICAT L'ARXIU «mitjana.c», POSTERIORMENT L'AFEGEIXO A L'ÀREA DE PREPARACIÓ I TORNO A VEURE L'ESTAT.

n) Torna a modificar el fitxer **mitjana.c**. A la línia 5, canvia **float mitjana;** per **float mitjana=0.0;**. Comprova l'estat del dipòsit.

```
romasarca@archlinux ~/mitjana (main)> vim mitjana.c
romasarca@archlinux ~/mitjana (main)> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   mitjana.c

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:   mitjana.c

romasarca@archlinux ~/mitjana (main)> █
```

Quina diferència hi ha entre els 2 fitxers **mitjana.c** mostrats per l'ordre?. Hi ha cap diferència entre l'àrea de preparació i el directori de treball?. Hi ha cap diferència entre l'àrea de preparació i el dipòsit?. Hi ha cap diferència entre el dipòsit i el directori de treball?.

ARA HI HA DUES VERSIONS DE «mitjana.c», LA DE COLOR VERD ÉS LA QUE TÉ «float = suma=0», MENTRES QUE LA DE COLOR VERMELL ÉS LA DARRERA VERSIÓ, LA QUE TÉ «float = suma=0.0» I ESTÀ EN EL DIRECTORI DE TREBALL. AMB LA QUAL COSA, AL DIPÒSIT (REPOSITORI) DE GIT SI ARA ES FÉS UN «git push» NO ES PUJARIA LA DARRERA VERSIÓ.

LES DIFERÈNCIES ES PODEN VEURE REALITZANT UN «git diff», PERÒ EN LA CAPTURA NO S'APRÈCIA, AIXÒ A CAUSA DE QUE EN EL MOMENT DE REALITZAR LA CAPTURA NO HI VAIG FER. LES OPCIONS PER VEURE LES DIFERÈNCIES SÓN LES SEGÜENTS:

«git diff» → INDICA DIFERÈNCIA ENTRE EL DIRECTORI DE TREBALL I L'ÀREA DE PREPARACIÓ.

«git diff --staged» → INDICA LA DIFERÈNCIA ENTRE L'ÀREA DE PREPARACIÓ I EL DARRER COMMIT.

«git diff main» → INDICA LA DIFERÈNCIA ENTRE EL DIRECTORI DE TREBALL I EL DARRER COMMIT DE LA BRANCA MAIN.

«git diff HEAD» → INDICA LA DIFERÈNCIA ENTRE EL DIRECTORI DE TREBALL I EL DARRER COMMIT (SEMPRE I QUAN EL HEAD NO HAGI ESTAT MOGUT A UN ALTRE COMMIT).

o) Fes un **add** del fitxer **mitjana.c** i comprova el seu estat. Quin fitxer hi ha a l'àrea de preparació?. Hi ha cap diferència entre l'àrea de preparació i el directori de treball?. Hi ha cap diferència entre l'àrea de preparació i el dipòsit?. Hi ha cap diferència entre el dipòsit i el directori de treball?.

```
romasarca@archlinux ~/mitjana (main)> git add mitjana.c
romasarca@archlinux ~/mitjana (main)> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   mitjana.c

romasarca@archlinux ~/mitjana (main)> █
```

p) Fes un **commit**. En el moment de consignar afegeix el comentari "*Tercer commit del programa mitjana*". Comprova:

- L'estat del projecte i el nom de la branca amb la qual estàs treballant
- L'historial.
- Hi ha cap diferència entre el directori de treball, l'àrea de preparació i el dipòsit?. Per què?

```
romasarca@archlinux ~/mitjana (main)> git commit -m "Tercer commit del programa mitjana"
[main 489ef73] Tercer commit del programa mitjana
 1 file changed, 2 insertions(+), 2 deletions(-)
romasarca@archlinux ~/mitjana (main)> git status
On branch main
nothing to commit, working tree clean
romasarca@archlinux ~/mitjana (main)> git log
commit 489ef7304d980235b4e4a6e7bba309484a15bb0f (HEAD -> main)
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Sun Oct 1 15:21:19 2023 +0000

    Tercer commit del programa mitjana

commit dd5873294e66ba9f9404d62874ab40df2c9febbe
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Wed Sep 27 17:54:23 2023 +0000

    Segon commit del projecte mitjana

commit 9fff719743e2748d043b39894fb25460709b8bf9
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Wed Sep 27 17:45:45 2023 +0000

    Primer commit del projecte mitjana
romasarca@archlinux ~/mitjana (main)> █
```

EN AQUEST CAS, A L'ÀREA DE PREPARACIÓ NO HI HA CAP DIFERÈNCIA, ENTTRE L'ÀREA DE PREPARACIÓ, SI ES FÉS UN «git diff», «git diff --staged» O «git diff HEAD», NO ENS HAURIA D'APARÈIXER RES.

5- Gestionant el sistema de control de versions des de l'interpret d'ordres per treballar amb dipòsits remots - Documentació (4)

a) Crea un projecte **Públic** de nom **mitjana** al teu dipòsit personal de Github. Veuràs que en el moment de la creació del nou dipòsit et donen informació de com utilitzar-lo. Llegeix la segona recomanació (..or push an existing repository from the command line) i fes un **push** del teu projecte local al dipòsit remot. Recorda que et demanarà com a password el token que vas crear a la primera pràctica.

```
romasarca@archlinux ~/mitjana (main)> git remote add origin https://github.com/romsar23daw2/mitjana.git
romasarca@archlinux ~/mitjana (main)> git branch -M main

romasarca@archlinux ~/mitjana (main)> git push -u origin main
Username for 'https://github.com': romsar23daw2
Password for 'https://romsar23daw2@github.com':
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.17 KiB | 1.17 MiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/romsar23daw2/mitjana.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
romasarca@archlinux ~/mitjana (main)> █
```

COM ES POT APRECIAR, HE SEGUIT LES INSTRUCCIONS CORRESPONENTS QUE INDICA GIT, PER AIXÍ PODER PUJAR UN REPOSITORI JA EXISTENT I POC MÉS. LO QUE ESTIC FENT ÉS PUJAR LA BRANCA MAIN LOCAL AL REPOSITORI DE GITHUB.

b) Crea una nova carpeta de nom **clonacions** dins del teu directori personal. **Clona** (Ajut: <https://git-scm.com/docs/git-clone>) des de dins de **clonacions** el projecte **mitjana**. Comprova que s'ha creat la carpeta **mitjana** i que dins d'aquesta carpeta hi ha un nou dipòsit local. Comprova l'estat del dipòsit clonat. Comprova l'historial. Comprova els dipòsits remots.

ARA LO QUE HE FET HA ESTAT CLONAR EL REPOSITORI QUE VAIG CREAR EN EL SEU TEMPS (27 DE SETEMBRE DE 2023). POSTERIORMENT MOSTRO L'HISTORIAL DEL REPOSITORI CLONAT.

```
romasarc@archlinux ~$ mkdir clonacions
romasarc@archlinux ~$ cd clonacions/
romasarc@archlinux ~$ git clone https://github.com/romsar23daw2/mitjana.git
Cloning into 'mitjana'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 2), reused 10 (delta 2), pack-reused 0
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (2/2), done.
romasarc@archlinux ~$ cd clonacions/
romasarc@archlinux ~$ ls
mitjana
romasarc@archlinux ~$ cd mitjana/
romasarc@archlinux ~$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
romasarc@archlinux ~$ git log
commit 489ef7304d980235b4e467bba309484a15bb0f (HEAD -> main, origin/main, origin/HEAD)
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Sun Oct 1 15:21:19 2023 +0000

    Tercer commit del programa mitjana

commit dd5873294e66ba9f9404d62874ab40df2c9febbe
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Wed Sep 27 17:54:23 2023 +0000

    Segon commit del projecte mitjana

commit 9fff719743e2748d043b39894fb25460709b8bf9
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Wed Sep 27 17:45:45 2023 +0000

    Primer commit del projecte mitjana
romasarc@archlinux ~$
```

c) Modifica el fitxer **mitjana.c** que es troba a **~/clonacions/mitjana**. A la línia 6, canvia **float nou_valor**; per **float nou_valor=0.0**; Fes un **add** i un **commit**. En el moment de consignar afegeix el comentari “*Quart commit del programa mitjana*”. Puja el canvi (fes un **push**) al dipòsit remot. . Comprova des de **Github** l'estat del dipòsit remot.

```
romasarc@archlinux ~$ cd mitjana/
romasarc@archlinux ~$ vim mitjana.c
romasarc@archlinux ~$ git add mitjana.c
romasarc@archlinux ~$ git commit -m "Quart comit del programa mitjana"
[main 00ccb9a] Quart comit del programa mitjana
1 file changed, 1 insertion(+), 1 deletion(-)
romasarc@archlinux ~$ git push
Username for 'https://github.com': romsar23daw2
Password for 'https://github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 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.
To https://github.com/romsar23daw2/mitjana.git
 489ef73..00ccb9a main -> main
romasarc@archlinux ~$ git log
commit 00ccb9a51e2ae2a555feef462b254c0b1657e9 (HEAD -> main, origin/main, origin/HEAD)
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Sun Oct 1 15:30:52 2023 +0000

    Quart comit del programa mitjana

commit 489ef7304d980235b4e467bba309484a15bb0f
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Sun Oct 1 15:21:19 2023 +0000

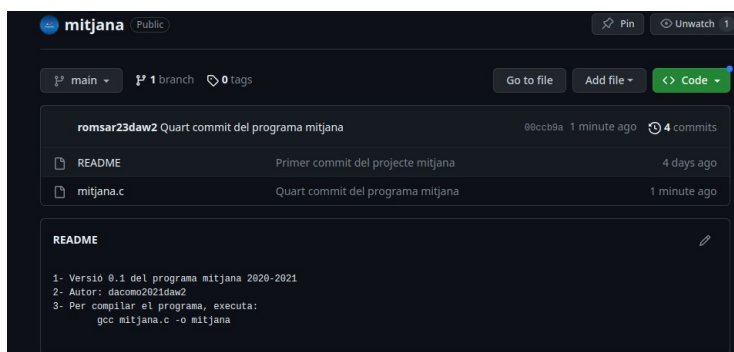
    Tercer commit del programa mitjana

commit dd5873294e66ba9f9404d62874ab40df2c9febbe
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Wed Sep 27 17:54:23 2023 +0000

    Segon commit del projecte mitjana

commit 9fff719743e2748d043b39894fb25460709b8bf9
Author: romsar23daw2 <15586186.clot@fje.edu>
Date:   Wed Sep 27 17:45:45 2023 +0000

    Primer commit del projecte mitjana
romasarc@archlinux ~$
```



ARA EL QUE HE FET HA ESTAT MODIFICAR LA LÍNIA 6, I POSTERIORMENT HE REALITZAT TOTS ELS PASSOS QUE S'HAVIEN DE FER («git add mitjana.c», «git commit -m «Quart comit del programa mitjana» I PER ACABAR, «git push»»), DESPRÉS HE COMPROVAT QUE L'ARXIU HA ESTAT MODIFICAT EN EL PROPI GITHUB. PER VEURE L'URL DEL REPOSITORI S'HA DE FER «git remote -v».

d) Torna a `~/mitjana` (el directori original). Baixa (fes un **pull**) del projecte que està al dipòsit remot al dipòsit local. Ara obre el fitxer `mitjana.c` i comprova que s'ha afegit el nou codi. **Ajut:** <https://try.github.io/levels/1/challenges/12>.

```
romasarca@archlinux ~/c/mitjana (main)> ~/mitjana/
romasarca@archlinux ~/mitjana (main)> git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), 302 bytes | 302.00 KiB/s, done.
From https://github.com/romsar23daw2/mitjana
 489ef73..00ccb9a  main    -> origin/main
Updating 489ef73..00ccb9a
Fast-forward
 mitjana.c | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
romasarca@archlinux ~/mitjana (main)> grep "float nou_valor" mitjana.c
float nou_valor=0.0;
romasarca@archlinux ~/mitjana (main)>
```

ARA HE CANVIAT DE CARPETA A `~/mitjana`, I POSTERIORMENT HE FET UN «git pull» PER PODER BAIXAR L'ÚLTIMA VERSIÓ DEL REPOSITORI.

6- Treballant en local: comprovant diferències, esborrants adds i commits, accedint a diferents versions.

a) Clona dins del teu directory personal el dipòsit <https://github.com/dacomo2021daw2/m08uf2pr1.git>. Mostra el log de commits amb l'opció `--pretty=oneline`.

```
romasarca@archlinux -> git clone https://github.com/dacomo2021daw2/m08uf2pr1.git
Cloning into 'm08uf2pr1'...
remote: Enumerating objects: 65, done.
remote: Counting objects: 100% (65/65), done.
remote: Compressing objects: 100% (63/63), done.
remote: Total 65 (delta 36), reused 2 (delta 0), pack-reused 0
Receiving objects: 100% (65/65), 14.98 KiB | 7.49 MiB/s, done.
Resolving deltas: 100% (36/36), done.
romasarca@archlinux -> m08uf2pr1/
romasarca@archlinux ~/m08uf2pr1 (main)> git log --pretty=oneline
7fa86f61da66db0b31ed0cb73c9ef5c34f62680c (HEAD -> main, origin/main, origin/HEAD) Update Vagrantfile
0fca4628b33799b749c35ae96a2891c71dd77d8 Update README
e6c638f519e73b08e29eb5e481ca48782c2705 Update .gitignore
26187d79056ad8e47b7b35e4b46543d912401c8a Update LICENSE
59bb79a8aacf1f6ded81e53441cb71ed008c9467 Update README
016d7fc4ff5ed87e0db2b783a70c5fcc47d0d2e Update README
d5c9021b477c50ff755c5da35a48cc58dbdb56a Update README
7c978124ef7473449a3efe892b85ff14e44684e1 Update README
da5a0d89833e974b4f73b3d692ffb07617be3506 Update Vagrantfile
2be6ca23f40d96192eb49ea6801eeb09b7cc6cf Update README
7c39f276b3420784a0aa3839aeb907c7feb70005 Update README
39a0db01581f0b35395f97474f2a3a3b70558a9b Update README
98f0cc0d6d03736a7379b5317421422b54f347 Update Vagrantfile
77ea17919b17f213d7f3ae11a9f02db0f01f479f Update Vagrantfile
30110da17a7eded4df0f0f57d2adbe9a1aa93f23 Update README
75d656e5907a932b57ee03dc714a2364254ee54e Update README
966ff8ed7511ca649417c3d348c0bbaf5b53a308 Update README
a22aa4174dc169fb3c91d655cd7be830eae62e9 Update Vagrantfile
b73a3177e16520a50d475ac4598aa47d7a4f217 Update README
0cddb622992e4d66be65b2b3ebd71fa96f56a98e Update README
31bb14382c088fb46840f881ccbf9ee4e245ca3e vagrant m08uf2pr2 daw2 2020-2021 FJE CLOT
romasarca@archlinux ~/m08uf2pr1 (main)>
```

EN AQUEST CAS, EL QUE FAIG ÉS CLONAR UN REPOSITORI, I MOSTRO L'HISTORIAL DE COMMITS PERÒ D'UNA FORMA MÉS COMPACTA.

SI FÉS ÚS DE L'OPCIÓ «git log --pretty=oneline | wc -l» MOSTRARIA EL NOMBRE TOTAL DE COMITS, QUE EN AQUEST CAS, ÉS 21.

b) Modifica el fitxer **README** que es troba al directori `m08uf1pr2`. Canvia a la **línia 1** i fes que el nou contingut sigui **# DAW2 M08UF4 PRÀCTICA 2 CURS 2023-2024 FJE CLOT**. Afegeix el fitxer **README** a l'àrea de preparació. Modifica també a la **línia 5** i fes que el nou contingut sigui **Requereix:**.

```
romasarca@archlinux ~/m08uf2pr1 (main)> vim README
romasarca@archlinux ~/m08uf2pr1 (main)> git add README
romasarca@archlinux ~/m08uf2pr1 (main)> vim README
```

ARA HE MODIFICAT L'ARXIU README, POSTERIORMENT L'HE AFEGIT A L'ÀREA DE TREBALL I MODIFICAT PERS SEGON COP.

c) Comprova:

- L'Estat del projecte
- La diferència entre el **directori de treball** i l'àrea de preparació
- La diferència entre el **directori de treball** i l'últim **commit**
- La diferència entre l'àrea de preparació i l'últim **commit**.

```
romasarca@archlinux ~/m08uf2pr1 (main)> vim README
romasarca@archlinux ~/m08uf2pr1 (main)> git add README
romasarca@archlinux ~/m08uf2pr1 (main)> vim README
romasarca@archlinux ~/m08uf2pr1 (main)> git add README
romasarca@archlinux ~/m08uf2pr1 (main)> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README

romasarca@archlinux ~/m08uf2pr1 (main)> git diff HEAD
diff --git a/README b/README
index 27405f3..5960a93 100644
--- a/README
+++ b/README
@@ -1,8 +1,8 @@
-# DAW2 M08UF1 2021-2022 FJE CLOT
+# DAW2 M08UF4 PRÀCTICA 2 CURS 2023-2024 FJE CLOT

## Requisites

-Requereix tenir instal·lat previament:
+Requereix:
1. Git https://git-scm.com/
2. Virtualbox https://www.virtualbox.org/
3. Vagrant https://www.vagrantup.com/
romasarca@archlinux ~/m08uf2pr1 (main)>
```

```
romasarca@archlinux ~/m08uf2pr1 (main)> git diff
romasarca@archlinux ~/m08uf2pr1 (main)>
```

```
romasarca@archlinux ~/m08uf2pr1 (main)> git diff --staged
diff --git a/README b/README
index 27405f3..5960a93 100644
--- a/README
+++ b/README
@@ -1,8 +1,8 @@
-# DAW2 M08UF1 2021-2022 FJE CLOT
+# DAW2 M08UF4 PRÀCTICA 2 CURS 2023-2024 FJE CLOT

## Requisites

-Requereix tenir instal·lat previament:
+Requereix:
1. Git https://git-scm.com/
2. Virtualbox https://www.virtualbox.org/
3. Vagrant https://www.vagrantup.com/
romasarca@archlinux ~/m08uf2pr1 (main)>
```

O SIGUI, QUE ARA TENIM...

d) Esborra el fitxer **README** que hi ha a l'àrea de preparació després de l'últim **add** realitzat (Documentació 4 - Apartat 2.4 - Unstaging a staged file). Comprova l'estat del projecte.

```
romasarca@archlinux ~/m08uf2pr1 (main)> git reset
Unstaged changes after reset:
M   README
romasarca@archlinux ~/m08uf2pr1 (main)> git status
On branch main
Your branch is up to date with 'origin/main'.

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:   README

no changes added to commit (use "git add" and/or "git commit -a")
romasarca@archlinux ~/m08uf2pr1 (main)>
```

PER BORRAR EL QUE HI HA HA L'ÀREA DE PREPARACIÓ S'HA DE FER UN «git reset».

e) Comprova novament l'historial de commits realitzats fins aquest punt. Llegeix la [aquesta documentació](#). Ara esborra l'últim **commit** realitzat a la branca **main** sense perdre els canvis al directori de treball. Visualitza ara l'historial de commits realitzats i comprova que l'últim **commit** ha estat esborrat. Comprova que els darrers canvis a **README** no s'han perdut.

```
romasarca@archlinux ~/m08uf2pr1 (main)> git log --pretty=oneline
7fa86f61da66db0b31ed0cb73c9ef5c34f62680c (HEAD -> main, origin/main, origin/HEAD) Update Vagrantfile
0fca4628be33799b749c35ae96a2891c71dd27d8 Update README
e6c638f5159e73b00e29e8bc5481ca4878262705 Update .gitignore
26187d79056ad8647b7b35e4b46543d912491c8a Update LICENSE
59bb79a8aacf1f6ded81e53441cb71ed008c9467 Update README
016d7fc4ff5ed87e0d6b2b783a70c5fcc47d0d2e Update README
d5c9021b477c50ff755c5da35a48cc58dbdb56a Update README
7c978124ef7473449a3efe892b85ff14e44684e1 Update README
da5a0d89833e974b4f73b3d692ffb07617be3506 Update Vagrantfile
2be6ca23f40d96192eb49ea6801eeb09b7ccc6cf Update README
7c39f276b3420784a0aa3839aeb907c7feb70005 Update README
39a0db01561f0b35395f97474f2a3a3b70558a9b Update README
98f0cc6d6d3b7e36a7379b5317421422b54f347 Update Vagrantfile
77ea17919b17f213df73ae11a9f02dbd701f479f Update Vagrantfile
30110da17a7eded4df0f57d2adbe9a1aa93f23 Update README
75d656e5907a932b57ee03dc714a2364254ee54e Update README
966ff8ed7511ca649417c3d348c0bbaf5b53a308 Update README
a22aa417a4c169fb3c91d655c4d7be830eae62e9 Update Vagrantfile
b73a3177e16520a50d4745ac4598aa47d7a4f217 Update README
0cddb622992e4d66be5b2b3ebd71fa96f56a98e Update README
31bb14382c088fb46840f881ccbf9ee4e245ca3e vagrant m08uf1pr2 daw2 2020-2021 FJE CLOT
romasarca@archlinux ~/m08uf2pr1 (main)>
```

```
romasarca@archlinux ~/m08uf2pr1 (main)> git reset --soft HEAD~1
romasarca@archlinux ~/m08uf2pr1 (main)> git log --pretty=oneline
0fca4628be33799b749c35ae96a2891c71dd27d8 (HEAD -> main) Update README
e6c638f5159e73b00e29e8bc5481ca4878262705 Update .gitignore
26187d79056ad8647b7b35e4b46543d912491c8a Update LICENSE
59bb79a8aacf1f6ded81e53441cb71ed008c9467 Update README
016d7fc4ff5ed87e0d6b2b783a70c5fcc47d0d2e Update README
d5c9021b477c50ff755c5da35a48cc58dbdb56a Update README
7c978124ef7473449a3efe892b85ff14e44684e1 Update README
da5a0d89833e974b4f73b3d692ffb07617be3506 Update Vagrantfile
2be6ca23f40d96192eb49ea6801eeb09b7ccc6cf Update README
7c39f276b3420784a0aa3839aeb907c7feb70005 Update README
39a0db01561f0b35395f97474f2a3a3b70558a9b Update README
98f0cc6d6d3b7e36a7379b5317421422b54f347 Update Vagrantfile
77ea17919b17f213df73ae11a9f02dbd701f479f Update Vagrantfile
30110da17a7eded4df0f57d2adbe9a1aa93f23 Update README
75d656e5907a932b57ee03dc714a2364254ee54e Update README
966ff8ed7511ca649417c3d348c0bbaf5b53a308 Update README
a22aa417a4c169fb3c91d655c4d7be830eae62e9 Update Vagrantfile
b73a3177e16520a50d4745ac4598aa47d7a4f217 Update README
0cddb622992e4d66be5b2b3ebd71fa96f56a98e Update README
31bb14382c088fb46840f881ccbf9ee4e245ca3e vagrant m08uf1pr2 daw2 2020-2021 FJE CLOT
romasarca@archlinux ~/m08uf2pr1 (main)>
```

PER ANAR A UN COMMIT CAP ENRERE S'HA DE FER «git reset --soft HEAD~1», COM ES POT VEURE A LA SEGONA CAPTURA, ARA EL HASH COMENÇA AMB «0fca», MENTRES QUE A LA PRIMERA (L'ORIGINAL) TÉ UN INICI AMB «7fa8».

f) Fes un **add** i un **commit** de **README** amb el comentari “Update README”. Comprova l'estat i el log del projecte. S'ha recuperat la connexió amb **origin**?

```
romasarca@archlinux ~/m08uf2pr1 (main)> git add README
romasarca@archlinux ~/m08uf2pr1 (main)> git commit -m "Update README"
[main 74511ef] Update README
2 files changed, 3 insertions(+), 3 deletions(-)
romasarca@archlinux ~/m08uf2pr1 (main)> git diff HEAD
romasarca@archlinux ~/m08uf2pr1 (main)> git log --pretty=oneline
74511efba4972b6320407b3842a16fd13cb8ded8 (HEAD -> main) Update README
0fca4628be33799b749c35ae96a2891c71dd27d8 Update README
e6c638f5159e73b00e29e8bc5481ca4878262705 Update .gitignore
26187d79056ad8647b7b35e4b46543d912491c8a Update LICENSE
59bb79a8aacf1f6ded81e53441cb71ed008c9467 Update README
016d7fc4ff5ed87e0d6b2b783a70c5fcc47d0d2e Update README
d5c9021b477c50ff755c5da35a48cc58dbdb56a Update README
7c978124ef7473449a3efe892b85ff14e44684e1 Update README
da5a0d89833e974b4f73b3d692ffb07617be3506 Update Vagrantfile
2be6ca23f40d96192eb49ea6801eeb09b7ccc6cf Update README
7c39f276b3420784a0aa3839aeb907c7feb70005 Update README
39a0db01561f0b35395f97474f2a3a3b70558a9b Update README
98f0cc6d6d3b7e36a7379b5317421422b54f347 Update Vagrantfile
77ea17919b17f213df73ae11a9f02dbd701f479f Update Vagrantfile
30110da17a7eded4df0f57d2adbe9a1aa93f23 Update README
75d656e5907a932b57ee03dc714a2364254ee54e Update README
966ff8ed7511ca649417c3d348c0bbaf5b53a308 Update README
a22aa417a4c169fb3c91d655c4d7be830eae62e9 Update Vagrantfile
b73a3177e16520a50d4745ac4598aa47d7a4f217 Update README
0cddb622992e4d66be5b2b3ebd71fa96f56a98e Update README
31bb14382c088fb46840f881ccbf9ee4e245ca3e vagrant m08uf1pr2 daw2 2020-2021 FJE CLOT
romasarca@archlinux ~/m08uf2pr1 (main)>
```

COM ES POT APRECIAR, SI FAIG UN «git diff HEAD», DIU QUE EL REPOSITORI NO TÉ CAP DIFERÈNCIA ENTRE EL DIRECTORI DE TREBALL.

g) Llegeix [aquesta documentació](#). Ara esborra l'últim **commit** realitzat a la branca **main** i en la mateixa operació, actualitza també els fitxers del directori de treball perquè siguin els mateixos que hi ha ara al darrer **commit**. Comprova l'estat i el log del projecte. Comprova l'estat del fitxer **README**.

```
romasarca@archlinux ~/m08uf2pr1 (main)> git reset --hard HEAD~1
HEAD is now at 0fca462 Update README
romasarca@archlinux ~/m08uf2pr1 (main)> git diff HEAD
romasarca@archlinux ~/m08uf2pr1 (main)> git log --pretty=oneline
0fca4628be33799b749c35ae96a2891c71dd27d8 (HEAD -> main) Update README
eec638f5159e73b00e29e8bc5481ca4878262705 Update .gitignore
26187d79056ad8647b7b35e4b46543d912491c8a Update LICENSE
59bb79a8aacf1f6ded81e53441cb71ed008c9467 Update README
016d7fc4ff5ed87e0d6b2b783a70c5fcc47d0d2e Update README
d5c9021b477c50ff755c5da35a48cc58dbdab56a Update README
7c978124ef7473449a3efe892b85ff14e44684e1 Update README
da5a0d89833e974b4f73b3d692ffb07617be3506 Update Vagrantfile
2be6ca23f40d96192eb49ea6801eeb09b7ccc6cf Update README
7c39f276b3420784a0aa3839aeb907c7feb70005 Update README
39a0db01561f0b35395f97474f2a3a3b70558a9b Update README
98f0cc60d6d3b7e36a7379b5317421422b54f347 Update Vagrantfile
77ea17919b17f213df73ae11a9f02dbd701f479f Update Vagrantfile
30110da17a7eded4df0f57d2adbe9a1aa93f23 Update README
75d656e5907a932b57ee03dc714a2364254ee54e Update README
966ff8ed7511ca649417c3d348c0bbaf5b53a308 Update README
a22aa417a4c169fb3c91d655c4d7be830eae62e9 Update Vagrantfile
b73a3177e16520a50d4745ac4598aa47d7a4f217 Update README
0cddb622992e4d66be65b2b3ebd71fa96f56a98e Update README
31bb14382c088fb46840f881ccb9ee4e245ca3e vagrant m08uf1pr2 daw2 2020-2021 FJE CLOT
romasarca@archlinux ~/m08uf2pr1 (main)>
```

ARA PER ESBORRAR L'ÚLTIM COMMIT DE LA BRANCA main, HE FET SERVIR L'OPCIÓ «--hard»

h) Comprova el contingut del fitxer **Vagrantfile**. Llegeix [aquesta documentació](#). Ara fes que aquest fitxer tingui el contingut que tenia en el primer commit del dipòsit. Comprova ara el seu contingut. Quines diferències hi ha amb l'original?.

```
romasarca@archlinux ~/m08uf2pr1 (main)> la
Permissions Size User Date Modified Name
drwxr-xr-x - romasarca 15 Nov 18:11 .git
-rw-r--r-- 16 romasarca 15 Nov 18:11 .gitignore
-rw-r--r-- 1.1k romasarca 15 Nov 18:11 LICENSE
-rw-r--r-- 2.1k romasarca 15 Nov 18:11 README
-rw-r--r-- 774 romasarca 15 Nov 18:11 Vagrantfile
romasarca@archlinux ~/m08uf2pr1 (main)> git log --pretty=oneline
0fca4628be33799b749c35ae96a2891c71dd27d8 (HEAD -> main) Update README
eec638f5159e73b00e29e8bc5481ca4878262705 Update .gitignore
26187d79056ad8647b7b35e4b46543d912491c8a Update LICENSE
59bb79a8aacf1f6ded81e53441cb71ed008c9467 Update README
016d7fc4ff5ed87e0d6b2b783a70c5fcc47d0d2e Update README
d5c9021b477c50ff755c5da35a48cc58dbdab56a Update README
7c978124ef7473449a3efe892b85ff14e44684e1 Update README
da5a0d89833e974b4f73b3d692ffb07617be3506 Update Vagrantfile
2be6ca23f40d96192eb49ea6801eeb09b7ccc6cf Update README
7c39f276b3420784a0aa3839aeb907c7feb70005 Update README
39a0db01561f0b35395f97474f2a3a3b70558a9b Update README
98f0cc60d6d3b7e36a7379b5317421422b54f347 Update Vagrantfile
77ea17919b17f213df73ae11a9f02dbd701f479f Update Vagrantfile
30110da17a7eded4df0f57d2adbe9a1aa93f23 Update README
75d656e5907a932b57ee03dc714a2364254ee54e Update README
966ff8ed7511ca649417c3d348c0bbaf5b53a308 Update README
a22aa417a4c169fb3c91d655c4d7be830eae62e9 Update Vagrantfile
b73a3177e16520a50d4745ac4598aa47d7a4f217 Update README
0cddb622992e4d66be65b2b3ebd71fa96f56a98e Update README
31bb14382c088fb46840f881ccb9ee4e245ca3e vagrant m08uf1pr2 daw2 2020-2021 FJE CLOT
romasarca@archlinux ~/m08uf2pr1 (main)> git checkout 31bb14 Vagrantfile
Updated 1 path from a23d4a8
romasarca@archlinux ~/m08uf2pr1 (main)> la Vagrantfile
Permissions Size User Date Modified Name
-rw-r--r-- 736 romasarca 15 Nov 18:12 Vagrantfile
romasarca@archlinux ~/m08uf2pr1 (main)> git checkout main
M
Vagrantfile
Already on 'main'
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
romasarca@archlinux ~/m08uf2pr1 (main)> la Vagrantfile
Permissions Size User Date Modified Name
-rw-r--r-- 736 romasarca 15 Nov 18:12 Vagrantfile
romasarca@archlinux ~/m08uf2pr1 (main)> git checkout main Vagrantfile
Updated 1 path from 8fca3ae
romasarca@archlinux ~/m08uf2pr1 (main)>
```

COM ES POT VEURE, L'ARXIU A L'INICI EL Vagrantfile PESA 774 BYTES, MIRO L'HISTORIAL DE COMMITS (I COPIO ELS PRIMERS 6 DÍGITS DEL DARRER COMMIT), AIXÒ PER PODER FER UN «git

**checkout [codi hash]», AIXÍ PODENT CANVIAR LA VERSIÓ DE L'ARXIU. PER ACABAR, TORNO AL L'ÚLTIMA VERSIÓ DE L'ARXIU AIXÍ ANANT A main.
PER ACABAR, MOSTRO EL TAMANY DE L'ARXIU, I COM ES POT VEURE EL TAMANY DE L'ARXIU ÉS MENOR.**

i) Fes que Vagrantfile torni a ser el del darrer commit utilitzant HEAD en comptes del codi hash. Comprova ara que el seu contingut és el del darrer commit.

```
romasarca@archlinux ~/m08uf2pr1 (main)> la
Permissions Size User Date Modified Name
drwxr-xr-x - romasarca 16 Nov 18:25 .git
-rw-r--r-- 16 romasarca 15 Nov 18:11 .gitignore
-rw-r--r-- 1.1k romasarca 15 Nov 18:11 LICENSE
-rw-r--r-- 2.1k romasarca 15 Nov 18:11 README
-rw-r--r-- 774 romasarca 16 Nov 18:25 Vagrantfile
romasarca@archlinux ~/m08uf2pr1 (main)> git checkout 31bb143 Vagrantfile
Updated 1 path from a23d4a8
romasarca@archlinux ~/m08uf2pr1 (main)> la Vagrantfile
Permissions Size User Date Modified Name
-rw-r--r-- 736 romasarca 16 Nov 18:25 Vagrantfile
romasarca@archlinux ~/m08uf2pr1 (main)> git checkout HEAD Vagrantfile
Updated 1 path from 8fcfa3e
romasarca@archlinux ~/m08uf2pr1 (main)> la Vagrantfile
Permissions Size User Date Modified Name
-rw-r--r-- 774 romasarca 16 Nov 18:26 Vagrantfile
romasarca@archlinux ~/m08uf2pr1 (main)>
```

BÀSICAMENT ARA HE FET LO MATEIX QUE A L'EXERCICI ANTERIOR, PERÒ EN COMPTES DE FER UN «git checkout main», HO HE FET AMB head.

j) Comprova la mida dels fitxers README i Vagrantfile. Fes que HEAD apunti al 10è commit de main començant des del final (Documentació 5.d). Comprova el log. Comprova la mida dels fitxers README i Vagrantfile. Què ha passat?.

```
romasarca@archlinux ~/m08uf2pr1 (main)> la README Vagrantfile
Permissions Size User Date Modified Name
-rw-r--r-- 2.1k romasarca 15 Nov 18:11 README
-rw-r--r-- 774 romasarca 16 Nov 18:26 Vagrantfile
romasarca@archlinux ~/m08uf2pr1 (main)> git log --pretty=oneline
0fca4628be33799b749c35ae96a2891c71dd27d8 (HEAD -> main) Update README
e6c38f5159e73b00e29e8bc5481ca4878262705 Update .gitignore
26187d79056ad8647b7b35e4b6543d912491c8a Update LICENSE
59bb79a8aac1f6ded81e53441cb71ed008c9467 Update README
016d7fc4f75ed87e9d6b2b783a7bc5fcc47d002e Update README
d5c9021b477c59ff75c5da35a48cc58dbdb36a Update README
7c978124ef7473449a3efe892b85ff14e44684e1 Update README
da5a0d08933e974b4f73b3d692ffb07617be3506 Update Vagrantfile
2be6ca23f40d96192eb49ea6801eeb09b7ccc6cf Update README
7c39f276b3420784a0aa3839aeb907c7feb70005 Update README
39a0db01561f0b35395f97474f2a3a3b70558a9b Update README
98f6cc0d6d3b7e36a7379b5317421422b54f347 Update Vagrantfile
77ea17019b1f212df73ae11e9f02bd0701479f Update Vagrantfile
38110da17a7eded4df0f8f57d2adbe9a1aa93f23 Update README
75d656e5907a932b57ee03dc714a2364254ee54e Update README
966ff8ed7511ca649417c3d348c0bba7f5b53a308 Update README
a22aa417a4c169fb3c91d655c4d7be830eae62e9 Update Vagrantfile
b73a3177e16520a50d4745ac4598aa47d7a4f217 Update README
0c0bb622992e4d66be5b2b3ebd71fa9f56a98e Update README
31bb14382c089fb6040f881cbf9e4e24bca3e vagrant m08uf2pr2 daw2 2020-2021 FJE CLOT
romasarca@archlinux ~/m08uf2pr1 (main)> git checkout main~10
Note: switching to 'main~10'.

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 39a0db0 Update README
romasarca@archlinux ~/m08uf2pr1 ((39a0db01))> la README Vagrantfile
Permissions Size User Date Modified Name
-rw-r--r-- 1.4k romasarca 16 Nov 18:29 README
-rw-r--r-- 772 romasarca 16 Nov 18:29 Vagrantfile
romasarca@archlinux ~/m08uf2pr1 ((39a0db01))>
```

PER COMENÇAR, HE VIST EL TAMANY DELS ARXIS, POSTERIORMENT HE MOSTRAT EL L'HISTORIAL DE VERSIONS. POSTERIORMENT HE FET «git checkout main~10», I HE MOSTRAT QUE EL TAMANY DE L'ARXIU ÉS INFERIOR.

k) Torna a fer que **HEAD** apunti al darrer commit de **main**. Canvia el fitxer **LICENSE** i fes que a la línia l'any del **Copyright** sigui **2023**. Comprova que hi ha una diferència entre el directori de treball i el darrer commit.

```
romasarca@archlinux ~/m08uf2pr1 ((39a0db01))> git checkout main
Previous HEAD position was 39a0db0 Update README
Switched to branch 'main'
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
romasarca@archlinux ~/m08uf2pr1 (main)> vim LICENSE
romasarca@archlinux ~/m08uf2pr1 (main)> git diff
diff --git a/LICENSE b/LICENSE
index c974534..70bfe25 100644
--- a/LICENSE
+++ b/LICENSE
@@ -1,6 +1,6 @@
 MIT License

-Copyright (c) 2021 dacom2021daw2
+Copyright (c) 2023 dacom2021daw2

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
romasarca@archlinux ~/m08uf2pr1 (main)>
```

ARA HE ANAT A L'ÚLTIM COMMIT DE MAIN, AIXÍ FENT UN «git checkout main», HE MODIFICAT LA LÍNIA DEMANADA, PERÒ AMB 2023, AIXÒ A CAUSA DE SER EL CURS QUE ESTIC REALITZANT, I FAIG UN «git diff» PER AIXÍ PODER VEURE LES DIFERÈNCIES.

l) Fes que **LICENSE** torni a tenir el contingut del darrer commit del **main** descarregant el fitxer del dipòsit al directori de treball.

```
romasarca@archlinux ~/m08uf2pr1 (main)> git checkout main LICENSE
Updated 1 path from 8fcfa3e
romasarca@archlinux ~/m08uf2pr1 (main)> git diff
romasarca@archlinux ~/m08uf2pr1 (main)>
```

ARA PER ACABAR, HE ESCRIT «git checkout main LICENSE», PER AIXÍ PODER TORNAR A TENIR L'ÚLTIMA VERSIÓ DEL FITXER.

Lliurament de la pràctica

- 1- Treball individual.
- 2- El nom del fitxer ha de tenir el següent format **daw2_cognom_nom_m08uf4pr2.pdf**
- 3- Envieu la solució proposada per correu electrònic:
Adreça: **cf@collados.org**
Assumpte: **daw2_cognom_nom_m08uf4pr2**
- 4- La **data límit** de lliurament de la pràctica és el dia **29-10-23** a les **23.59 hores**.
- 5- Format Lletra: **Arial 10**. Marges (superior, inferior, esquerra, dreta): **2cm**