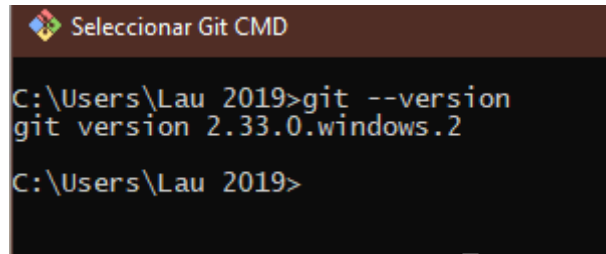


2. Fundamentos Git I

1.- Instalamos GIT y comprobamos la versión.

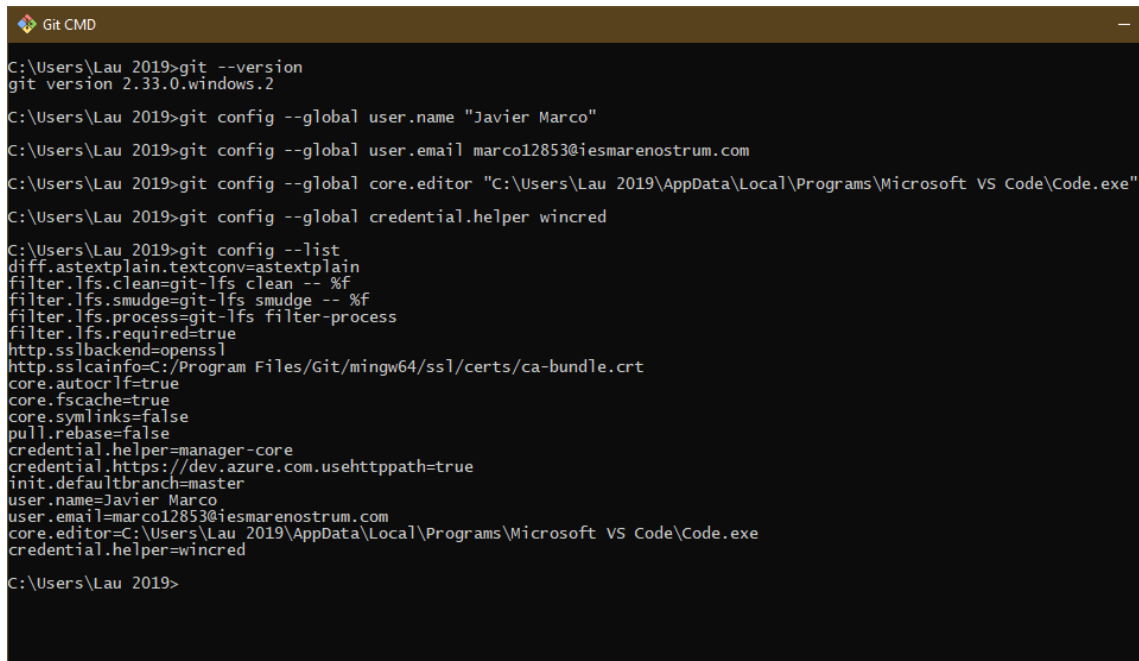


```
Selecccionar Git CMD

C:\Users\Lau 2019>git --version
git version 2.33.0.windows.2

C:\Users\Lau 2019>
```

2.- Realizamos la configuración básica de GIT.



```
Git CMD

C:\Users\Lau 2019>git --version
git version 2.33.0.windows.2

C:\Users\Lau 2019>git config --global user.name "Javier Marco"

C:\Users\Lau 2019>git config --global user.email marco12853@iesmarenostrum.com

C:\Users\Lau 2019>git config --global core.editor "C:\Users\Lau 2019\AppData\Local\Programs\Microsoft VS Code\Code.exe"

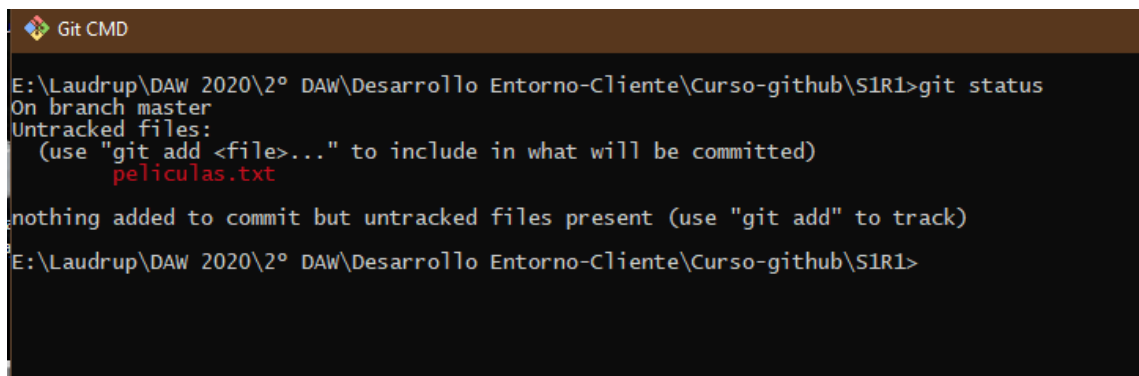
C:\Users\Lau 2019>git config --global credential.helper wincred

C:\Users\Lau 2019>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Javier Marco
user.email=marco12853@iesmarenostrum.com
core.editor=C:\Users\Lau 2019\AppData\Local\Programs\Microsoft VS Code\Code.exe
credential.helper=wincred

C:\Users\Lau 2019>
```

3.- Creamos una carpeta llamada S1R1.

3.g) Hacemos una captura mostrando el comando **git status**.



```
Git CMD

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    peliculas.txt

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

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>
```

Todavía no lo hemos añadido, ya que lo veremos más adelante junto con más archivos, por ello nos saldrá marcado en rojo.

3.i) Una vez que hemos creado los archivos que se nos ha solicitado (películas.txt y comidas.txt) los añadimos y hacemos el commit de la tercera versión

```
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git commit -m "Tercera versión"
[master 9d4e71d] Tercera versión
 2 files changed, 6 insertions(+)
 create mode 100644 comidas.txt
 create mode 100644 películas.txt
```

3.l) Llegados a este punto, habremos eliminado el archivo comidas y habremos modificado películas.

```
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:    comidas.txt
        modified:   películas.txt
```

3.n) Una vez añadidos al área de preparación, mostramos con **git status** como hemos eliminado el archivo comidas y hemos modificado el archivo películas.

```
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    comidas.txt
        modified:   películas.txt

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>
```

3.q) Hemos creado un archivo llamado **datos.bak** y una subcarpeta llamada **output** con otro archivo en su interior llamado **salida.txt**

```
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        datos.bak
        output/

nothing added to commit but untracked files present (use "git add" to track)
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>
```

Como podemos comprobar, no nos muestra el archivo dentro de la carpeta Output. Hacemos un **add .** para visualizarlo.

```

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        datos.bak
        output/

nothing added to commit but untracked files present (use "git add" to track)
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git add .
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   datos.bak
        new file:   output/salida.txt
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>

```

s) Una vez creado el archivo **.gitignore**, comprobamos como con **git status** nos sigue mostrando los archivos que hemos introducido anteriormente con **add** para que nos saliera la visualización correcta.

Es por ello que procedemos a sacarlos utilizando **git restore --staged <file>** (donde file será el nombre del archivo) y ahora sí, comprobamos como solamente tenemos **untracked** nuestro archivo **.gitignore**.

```

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   datos.bak
        new file:   output/salida.txt

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

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git restore --staged <file>
La sintaxis del comando no es correcta.

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git restore --staged <datos.bak>
La sintaxis del comando no es correcta.

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git restore --staged datos.bak
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git restore --staged output/salida.txt
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

```

u) Para terminar con esta parte de la práctica, hacemos un **git log** para mostrar el histórico de cambios y comprobar que están los **5 commits** que hemos realizado.

```
E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>git log
commit 7cb8280a5662bb9ccd8151c00de109a0aae8ed63 (HEAD -> master)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Wed Sep 22 13:11:48 2021 +0200

    Quinta versión

commit d5df47c05cd0538cee3a6c8df352e48ab0fae61d
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Wed Sep 22 12:46:42 2021 +0200

    Cuarta versión

commit 9d4e71d4749b51ed39237ff2d7b190d29c6f02be
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Wed Sep 22 12:35:32 2021 +0200

    Tercera versión

commit c2884d54c84e26924f235dc5d4aac0d10837a522
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Wed Sep 22 12:28:14 2021 +0200

    Segunda versión

commit ac99da1e6f345a031c728b45e6331ea0ff525667
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Wed Sep 22 12:11:31 2021 +0200

    Primera versión

E:\Laudrup\DAW 2020\2º DAW\Desarrollo Entorno-Cliente\Curso-github\S1R1>
```

Fundamentos Git II

Para la realización de esta parte de la práctica utilizaremos el Git Bash en vez de Git básico. Dado que el profesor ha comentado en su videotutorial que depende de gustos y de si preferimos una interfaz de comandos tipo Windows o Linux, vamos a probar Git Bash para ver con cual estamos más cómodos de cara a seguir trabajando en el futuro en la misma línea.

1. Creamos una carpeta denominada S2R1. Creamos el repositorio Git y añadimos un fichero con tres actores. Hacemos un primer commit. Crearemos una rama llamada test y cambiamos a esta nueva rama. Ahora añadimos un fichero con 3 actrices y hacemos un commit.

1.g) Hacemos una captura mostrando el comando **git log --graph --all**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (test)
$ git log --graph --all
* commit 71899ed579fedb9aa6fe8d607764d345a082919e (HEAD -> test)
  Author: Javier Marco <marco12853@iesmarenosttrum.com>
  Date:   Wed Sep 22 16:49:56 2021 +0200

    Segundo commit Actrices

* commit 40651bfac82a4099d9d2f20a37daf70d55de375e (master)
  Author: Javier Marco <marco12853@iesmarenosttrum.com>
  Date:   Wed Sep 22 16:43:05 2021 +0200

    Primera versión

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (test)
$
```

1.i) Ahora volvemos a la **rama master** y al ver que actrices todavía no existe en esta rama, tenemos que incorporar los cambios que hicimos de la rama test. Para ello, desde la **rama master** vamos a utilizar el comando **git merge <nombre de la rama final>**, en nuestro caso, **git merge test**.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (test)
$ git checkout master
Switched to branch 'master'

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master)
$ git merge test
Updating 40651bf..71899ed
Fast-forward
 actrices.txt | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 actrices.txt

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master)
$
```

Una vez que están en el mismo punto la rama master y la rama test, podemos borrar esa rama test que habíamos creado al principio. Como bien se menciona en el videotutorial, este paso es opcional pero dado que la práctica nos va a pedir de ahora en adelante la creación de otra rama llamada **test2**, vamos a probar cómo funciona el borrado de una rama. Vamos a utilizar el comando: **git branch -d <nombre de la rama a borrar>**. Nuestro caso; **git branch -d test**.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master)
$ git branch -d test
Deleted branch test (was 71899ed).
```

1.j) Después de crear la nueva rama, hacemos un **git log --graph --all** para comprobar que tanto nuestra rama master como nuestra nueva rama test2 están en el mismo punto y ya no existe la rama test1.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master)
$ git log --graph --all
* commit 71899ed579fedb9aa6fe8d607764d345a082919e (HEAD -> master, test2)
| Author: Javier Marco <marco12853@iesmarenostrum.com>
| Date: Wed Sep 22 16:49:56 2021 +0200
|
| Segundo commit Actrices
|
* commit 40651bfac82a4099d9d2f20a37daf70d55de375e
| Author: Javier Marco <marco12853@iesmarenostrum.com>
| Date: Wed Sep 22 16:43:05 2021 +0200
|
| Primera versión
|
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master)
$ |
```

1.n) Hemos añadido una actriz en la rama master y luego hemos cambiado a la rama test2 y hemos añadido otra actriz diferente. Mostramos en qué punto están nuestras ramas con: **git log --graph --all** y vemos que tenemos dos caminos distintos.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (test2)
$ git log --graph --all
* commit bd56e3e0497b786e026721149ad1b27657ca3052 (HEAD -> test2)
| Author: Javier Marco <marco12853@iesmarenostrum.com>
| Date: Wed Sep 22 17:15:52 2021 +0200
|
| Añadir Jennifer Aniston
|
* commit c861a5da31a5f98eee00afd74effbc8b8d25ca38 (master)
| Author: Javier Marco <marco12853@iesmarenostrum.com>
| Date: Wed Sep 22 17:14:20 2021 +0200
|
| Añadir Courteney Cox
|
* commit 71899ed579fedb9aa6fe8d607764d345a082919e
| Author: Javier Marco <marco12853@iesmarenostrum.com>
| Date: Wed Sep 22 16:49:56 2021 +0200
|
| Segundo commit Actrices
|
* commit 40651bfac82a4099d9d2f20a37daf70d55de375e
| Author: Javier Marco <marco12853@iesmarenostrum.com>
| Date: Wed Sep 22 16:43:05 2021 +0200
|
| Primera versión
|
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (test2)
$ |
```

1.o) Una vez que hemos cambiado a la rama master. Hemos intentado hacer **git merge test2** para incorporar los cambios como hicimos los pasos previos con la rama test1 pero esta vez nos produce un conflicto.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (test2)
$ git checkout master
Switched to branch 'master'

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master)
$ git merge test2
Auto-merging actrices.txt
CONFLICT (content): Merge conflict in actrices.txt
Automatic merge failed; fix conflicts and then commit the result.

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master|M
ERGING)
$ |
```

Esto se debe a que GIT detecta que hay cambios en ambas ramas y no es capaz de gestionar cuál de las dos ramas es la adecuada, por lo que necesitará nuestra ayuda para solucionarlo. Vamos a comprobar el estado con: **git status**.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master|M
ERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   actrices.txt

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

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master|M
ERGING)
$ |
```

1.q) Desde Visual Studio Code (el core.editor que estamos utilizando) nos muestra el conflicto y una serie de opciones. Vemos como en la rama master teníamos Courteney Cox y en la rama test2 tenemos Jennifer Aniston.

```
1  Uma Thurman
2  Scarlett Johansson
3  Gal Gadot
4  <<<<<< HEAD (Current Change)
5  Courteney Cox
6  =====
7  Jennifer Aniston
8  >>>>>> test2 (Incoming Change)
9
```

Podemos aceptar una de las dos ramas como la buena, aceptar ambos cambios y tenemos una última **opción de comparar**, la cual vamos a seleccionar para ver como lo muestra.

1	Uma Thurman	1	Uma Thurman
2	Scarlett Johansson	2	Scarlett Johansson
3	Gal Gadot	3	Gal Gadot
4-	Courteney Cox	4+	Jennifer Aniston
5		5	

Esta opción nos muestra en pantalla partida como se quedarían en caso de elegir una opción o la otra. En nuestro caso, como no son excluyentes una de la otra, vamos a seleccionar la opción: **Accept Both Changes**, de manera que nos combina las dos opciones.

1	Uma Thurman
2	Scarlett Johansson
3	Gal Gadot
4	Courteney Cox
5	Jennifer Aniston
6	

Esta opción viene dada por **Visual Studio Code**, con otros programas puede que también venga alguna opción integrada similar y si no, manualmente deberemos modificar las líneas eliminando las referencias que nos muestra el conflicto:

```
<<<<<< HEAD          ===== >>>>>> test2
```

1.q) Los añadimos, hacemos commit y comprobamos el resultado de las ramas otra vez con el comando: **git log --graph --all**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master|M
ERGING)
$ git commit -m "Conflicto actrices resuelto"
[master 5d493de] Conflicto actrices resuelto

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master)
$ git log --graph --all
* commit 5d493de3357fbac59df8ffdd4a96db734cce1756 (HEAD -> master)
  | Merge: c861a5d bd56e3e
  | Author: Javier Marco <marco12853@iesmarenostrum.com>
  | Date:   Wed Sep 22 17:38:51 2021 +0200
  |
  |     Conflicto actrices resuelto
  |
  | * commit bd56e3e0497b786e026721149ad1b27657ca3052 (test2)
  |   | Author: Javier Marco <marco12853@iesmarenostrum.com>
  |   | Date:   Wed Sep 22 17:15:52 2021 +0200
  |   |
  |   |     Añadir Jennifer Aniston
  |   |
  |   | * commit c861a5da31a5f98eee00afd74effbc8b8d25ca38
  |   |   | Author: Javier Marco <marco12853@iesmarenostrum.com>
  |   |   | Date:   Wed Sep 22 17:14:20 2021 +0200
  |   |   |
  |   |   |     Añadir Courteney Cox
  |   |   |
  |   |   | * commit 71899ed579fedb9aa6fe8d607764d345a082919e
  |   |   |   | Author: Javier Marco <marco12853@iesmarenostrum.com>
  |   |   |   | Date:   Wed Sep 22 16:49:56 2021 +0200
  |   |   |   |
  |   |   |   |     Segundo commit Actrices
  |   |   |   |
  |   |   |   | * commit 40651bfac82a4099d9d2f20a37daf70d55de375e
  |   |   |   |   | Author: Javier Marco <marco12853@iesmarenostrum.com>
  |   |   |   |   | Date:   Wed Sep 22 16:43:05 2021 +0200
  |   |   |   |   |
  |   |   |   |   |     Primera versión
  |   |   |   |   |
  |   |   |   |   | Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R1 (master)
  |   |   |   |   | $
```


2. Creamos una carpeta llamada S2R2-remoto y utilizaremos un repositorio **bare** (“pelado”, “vacio”).

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github
b/S2R2-remoto
$ git init --bare
Initialized empty Git repository in E:/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2-remoto/
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2-remoto (BARE:master)
$ |
```

3. Para realizar el clonado, primero volvemos a la carpeta de nivel anterior (comando **cd ..**). Una vez aquí, para clonar utilizaremos el comando: **git clone URL**. En nuestro caso como es una carpeta, ponemos la ruta de la carpeta; **git clone S2R2-remoto/** además, añadiremos el nombre con el cual lo queremos nombrar, dado que por defecto lo llamará con el mismo nombre. De esta forma, hemos utilizado el comando: **git clone S2R2-remoto/ S2R2**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github
$ git clone S2R2-remoto/ S2R2
Cloning into 'S2R2'...
warning: You appear to have cloned an empty repository.
done.
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github
$ |
```

Ahora, creamos un archivo llamado directores y añadimos 3 nombres de directores de cine. Hacemos un commit.

3.c) Antes de hacer **push**, comprobamos con el comando: **git status** como nuestra rama está basada en ‘Origin/master’. Ahora utilizaremos el comando: **git push** (opcionalmente podríamos añadir el nombre del remoto y la rama, es decir, **git push origin master**, pero ya sabemos que nuestra rama está basada en ello y no es preciso.)

```
create mode 100644 directores.txt
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 272 bytes | 272.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To E:/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2-remoto/
 * [new branch]      master -> master
```

Creamos una rama nueva denominada version1 y cambiamos a esta nueva rama.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git branch version1
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git checkout version1
Switched to branch 'version1'
```

Añadimos dos directores al archivo y hacemos un commit.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (version1)
$ git add .

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (version1)
$ git commit -m "Añadimos 2 directores"
[version1 b5da29e] Añadimos 2 directores
1 file changed, 3 insertions(+), 1 deletion(-)
```

3.g) Tal y como hicimos en los pasos anteriores, haremos un push de la rama al repositorio remoto. Para ello, utilizaremos el comando **git push** indicando cual es nuestra rama con el comando: **git push origin version1**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (version1)
$ git push origin version1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 340 bytes | 340.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To E:/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2-remoto/
* [new branch]      version1 -> version1
```

4. Clonamos el repositorio S2R2-remoto en una segunda carpeta llamada S2R3

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github
$ git clone S2R2-remoto/ S2R3
Cloning into 'S2R3'...
done.
```

4.a) Mostraremos el contenido de directores.txt. Para ello, he utilizado el comando **git log** para ver el último commit que hice.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git log
commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (HEAD -> master, origin/master, origin/HEAD)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date:   Mon Sep 27 16:24:28 2021 +0200

    Directores
```

Después, he hecho un **git show** con el nombre del commit. Observamos como tenemos los 3 directores que añadimos al principio de la práctica.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git show 7e978a0d
commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (HEAD -> master, origin/master, origin/HEAD)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date:   Mon Sep 27 16:24:28 2021 +0200

    Directores

diff --git a/directores.txt b/directores.txt
new file mode 100644
index 0000000..7a795c1
--- /dev/null
+++ b/directores.txt
@@ -0,0 +1,3 @@
+Denis Villeneuve
+Peter Jackson
+Robert Zemeckis
\ No newline at end of file
```

También podremos utilizar el comando **git diff** para visualizar el contenido.

Por último, mostramos **git status**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

4.b) Cambiamos a la rama version1 con; **git checkout version1**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git checkout version1
Switched to a new branch 'version1'
Branch 'version1' set up to track remote branch 'version1' from 'origin'.
```

Branch 'version1' set up to track remote branch 'version1' from 'origin'.

Podemos observar como el mensaje no es el mismo que cuando creamos la rama en S2R2, que nos mostraba *Switched to branch version1*.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git checkout version1
Switched to branch 'version1'
```

4.c) Ahora mostramos el contenido del fichero directores.txt y esta vez vemos como hemos añadido dos directores. (En nuestro caso, vemos que hemos añadido 3 y eliminamos 1 pero se debe a un error a la hora de escribir encima de la misma línea del archivo.)

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (version1)
$ git log
commit b5da29e526b9413c6e72c440525e11dba8f037f9 (HEAD -> version1, origin/version1)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 17:06:29 2021 +0200

    Añadimos 2 directores

commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (origin/master, origin/HEAD, master)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 16:24:28 2021 +0200

    Directores

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (version1)
$ git show b5da29e5
commit b5da29e526b9413c6e72c440525e11dba8f037f9 (HEAD -> version1, origin/version1)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 17:06:29 2021 +0200

    Añadimos 2 directores

diff --git a/directores.txt b/directores.txt
index 7a795c1..55c54e7 100644
--- a/directores.txt
+++ b/directores.txt
@@ -1,3 +1,5 @@
 Denis Villeneuve
 Peter Jackson
 -Robert Zemeckis
 \ No newline at end of file
+Robert Zemeckis
+Steven Spielberg
+Bong Joon-ho
 \ No newline at end of file
```

4.d) Cambiamos a la rama master; **git checkout master**.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (version1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

Comprobamos como al cambiar a la rama master nos informa: “Your branch is up to date with ‘origin/master’”.

Utilizamos el comando: **git log --all** y vemos que en la rama version1 tenemos el commit donde añadíamos los 2 directores extra.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git log --all
commit b5da29e526b9413c6e72c440525e11dba8f037f9 (origin/version1, version1)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 17:06:29 2021 +0200

    Añadimos 2 directores

commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (HEAD -> master, origin/master, origin/HEAD)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 16:24:28 2021 +0200

    Directores
```

Importante utilizar el comando **git log --all** para mostrar todos, de lo contrario, con **git log** solamente nos mostrará el commit de master.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git log
commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (HEAD -> master, origin/master, origin/HEAD)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 16:24:28 2021 +0200

    Directores
```

4.e) Incorporamos los cambios de la rama versión1 a la rama master con el comando: **git merge version1**.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git merge version1
Updating 7e978a0..b5da29e
Fast-forward
 directores.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

Comprobamos con un **git log --all**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git log --all
commit b5da29e526b9413c6e72c440525e11dba8f037f9 (HEAD -> master, origin/version1, version1)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 17:06:29 2021 +0200

    Añadimos 2 directores

commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (origin/master, origin/HEAD)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 16:24:28 2021 +0200

    Directores
```

Con esto vemos que hemos traído versión1 pero hemos dejado atrás origin/master.

Con un **git status** observamos el mensaje que nos advierte: “Your branch is ahead of ‘origin/master’ by 1 commit. (use “git push” to publish your local commits)”.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Tal y como nos dice el mensaje, hacemos **git push**. Después comprobamos de nuevo **git status**. Ahora mostramos **git log --all** y vemos que esta todo en sincronía.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To E:/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2-remoto/
  7e978a0..b5da29e  master -> master

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git log --all
commit b5da29e526b9413c6e72c440525e11dba8f037f9 (HEAD -> master, origin/version1, origin/master, origin/HEAD, version1)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date:   Mon Sep 27 17:06:29 2021 +0200

    Añadimos 2 directores

commit 7e978a0d5583114ac23a14da0b32eda8b3e05960
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date:   Mon Sep 27 16:24:28 2021 +0200

    Directores
```

4.f) Sube la rama master actualizada al servidor.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git pull
Already up to date.
```

Si hacemos un **git fetch**, no nos da ningún mensaje porque no hay nada nuevo.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ git fetch

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R3 (master)
$ |
```

5. Volvemos a la carpeta S2R2 y mostramos como está la situación gráfica con el comando ya conocido: **git log --graph --all**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git log --graph --all
* commit b5da29e526b9413c6e72c440525e11dba8f037f9 (origin/version1, version1)
  Author: Javier Marco <marco12853@iesmarenostrum.com>
  Date:   Mon Sep 27 17:06:29 2021 +0200

      Añadimos 2 directores

* commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (HEAD -> master, origin/master)
  Author: Javier Marco <marco12853@iesmarenostrum.com>
  Date:   Mon Sep 27 16:24:28 2021 +0200

      Directores
```

5.a) Realizamos un **git fetch**

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git fetch
From E:/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2-remoto
 7e978a0..b5da29e master -> origin/master

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$
```

Y volvemos a mostrar el gráfico.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git fetch
From E:/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2-remoto
 7e978a0..b5da29e master -> origin/master

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git log --graph --all
* commit b5da29e526b9413c6e72c440525e11dba8f037f9 (origin/version1, origin/master, version1)
  Author: Javier Marco <marco12853@iesmarenostrum.com>
  Date: Mon Sep 27 17:06:29 2021 +0200

    Añadimos 2 directores

* commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (HEAD -> master)
  Author: Javier Marco <marco12853@iesmarenostrum.com>
  Date: Mon Sep 27 16:24:28 2021 +0200

    Directores
```

Podemos observar cómo hemos traído el origin/master pero nosotros seguimos por detrás, por eso si hacemos un **git show** nos mostrará todavía los 3 directores en nuestra rama master.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git show
commit 7e978a0d5583114ac23a14da0b32eda8b3e05960 (HEAD -> master)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 16:24:28 2021 +0200

    Directores

diff --git a/directores.txt b/directores.txt
new file mode 100644
index 0000000..7a795c1
--- /dev/null
+++ b/directores.txt
@@ -0,0 +1,3 @@
+Denis Villeneuve
+Peter Jackson
+Robert Zemeckis
\ No newline at end of file
```

5.b) Ahora realizamos **git merge origin/master** y como es un camino directo, nos lo hace con un fast-forward, mezclando la rama remota con la rama actual para avanzarnos.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git merge origin/master
Updating 7e978a0..b5da29e
Fast-forward
 directores.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

(Recordemos que podríamos haber hecho el paso a y b juntos con un **git pull**)

5.c) Ahora hacemos **git show** para ver que ahora sí nos muestra los 5 directores.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/S2R2 (master)
$ git show
commit b5da29e526b9413c6e72c440525e11dba8f037f9 (HEAD -> master, origin/version1, origin/master, version1)
Author: Javier Marco <marco12853@iesmarenostrum.com>
Date: Mon Sep 27 17:06:29 2021 +0200

    Añadimos 2 directores

diff --git a/directores.txt b/directores.txt
index 7a795c1..55c54e7 100644
--- a/directores.txt
+++ b/directores.txt
@@ -1,3 +1,5 @@
+Denis Villeneuve
+Peter Jackson
-Robert Zemeckis
\ No newline at end of file
+Robert Zemeckis
+Steven Spielberg
+Bong Joon-ho
\ No newline at end of file
```

4. GitHub

Utilizaremos la cuenta de GitHub que teníamos en 1º, JMF1DAW1. Eso sí, hemos añadido el correo de educación de IES Mare Nostrum y solicitado el descuento educativo.

4.a) Tras hacer fork del repositorio, hacemos el clonado.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github
$ git clone https://github.com/JMF1DAW1/sesion3-practica.git
Cloning into 'sesion3-practica'...
remote: Enumerating objects: 10, done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 10
Receiving objects: 100% (10/10), done.
```

Una vez tenemos la carpeta, accedemos a ella y con el comando **ls** comprobamos los archivos que hay.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github
$ cd S
S1R1/          S2R1/          S2R2/          S2R2-remoto/   S2R3/          sesion3-practica/

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github
$ cd sesion3-practica/

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ ls
LICENSE README.md discos.txt libros.txt peliculas.txt

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ |
```


Realizamos el cambio al fichero README.md para añadir nuestro nombre y apellido. Comprobamos con el comando **git status** que nos marca que hemos modificado el archivo. Utilizamos los comandos **git add .**, seguido de **git commit -m "Nombre y apellidos"** y por último, subimos los cambios a nuestra copia del repositorio con **git push**.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git status
On branch master
Your branch is up to date with 'origin/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:   README.md

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

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git add .

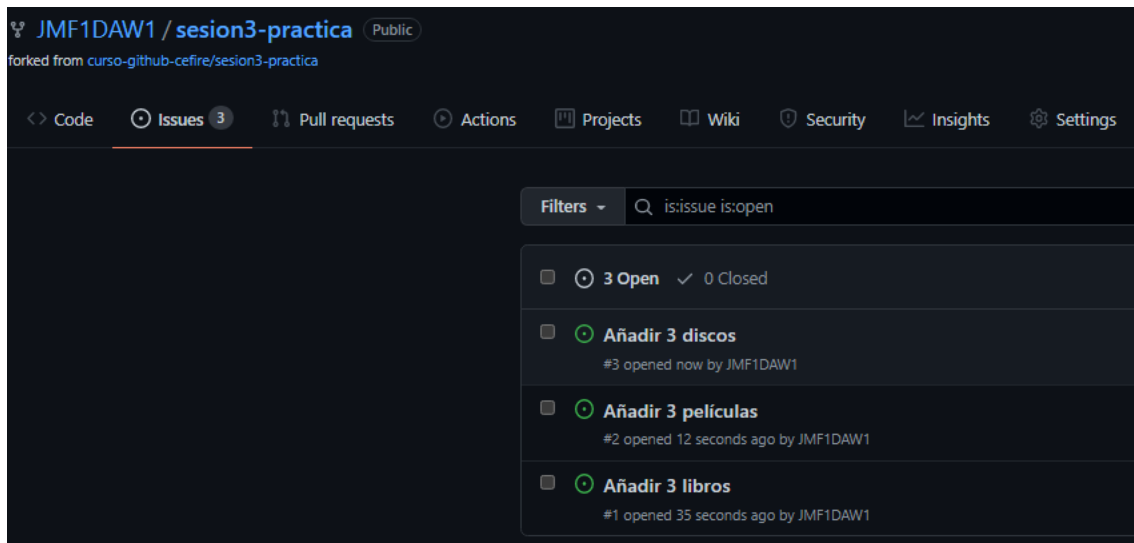
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git commit -m "Nombre y apellidos"
[master d49efb4] Nombre y apellidos
1 file changed, 2 insertions(+)

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/JMF1DAW1/sesion3-practica.git
   a692f8f..d49efb4  master -> master
```

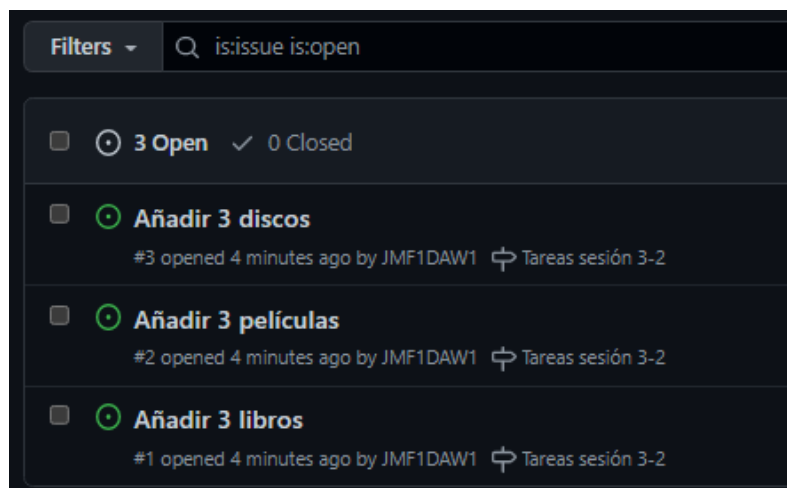
Ahora vemos como en nuestra cuenta de GitHub se ha modificado el archivo y además nos informa de que la rama va 1 commit por delante.



4.b) Creamos 3 **issues** con el nombre de las tres tareas que vamos a realizar.



c) Creamos un **milestone** y añadimos los 3 issues que hemos creado antes.



4.d) En este punto, deberíamos modificar cada fichero individualmente y subirlo para cerrar su respectivo Issue. Sin embargo, también lo podremos hacer todo junto y cerrar diferentes issues simultáneamente. Nosotros vamos a modificar los tres ficheros juntos y realizamos un **git status** para comprobarlo.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git status
On branch master
Your branch is up to date with 'origin/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:   discos.txt
        modified:   libros.txt
        modified:   películas.txt

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

Utilizamos el comando **git diff** para ver esos cambios. (Hemos sustituido los tres que venían en cada archivo por otros tres diferentes).

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git diff
diff --git a/discos.txt b/discos.txt
index 3594f9a..eff4f6b 100644
--- a/discos.txt
+++ b/discos.txt
@@ -1,3 +1,3 @@
-The Beatles
-Ella & Louis
-Jagged Little Pill
+Metora
+La primavera del invierno
+Infrasoinuak
diff --git a/libros.txt b/libros.txt
index d127caf..66a708e 100644
--- a/libros.txt
+++ b/libros.txt
@@ -1,3 +1,3 @@
-Juego de Tronos
-El Juego de Ender
-La Historia Interminable
+El señor de los anillos
+El nombre del viento
+El archivo de las tormentas
diff --git a/peliculas.txt b/peliculas.txt
index 29abf38..6515886 100644
--- a/peliculas.txt
+++ b/peliculas.txt
@@ -1,3 +1,3 @@
-Orguillo y Prejuicio
-La boda de mi mejor amiga
-Alta Fidelidad
+Dune
+Matrix
+El padrino
```

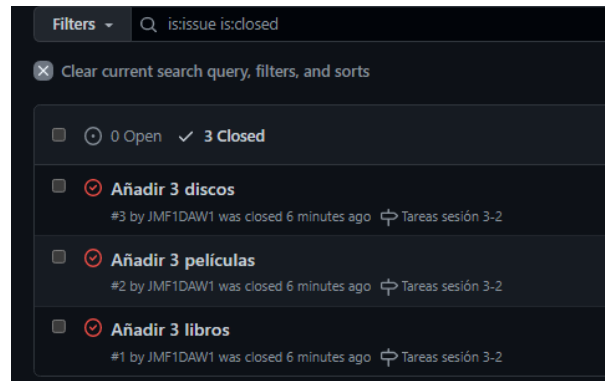
Ahora debemos hacer un commit pero con la peculiaridad de que tenemos que cerrar los issues de manera automática. Para ello, utilizaremos el comando: **git commit -m “Modificaciones. Close #1, Close #2, Close #3”**. Nosotros estamos cerrando los tres juntos. En caso de que hubiésemos modificado sólo el primer archivo, usaríamos Close #1 y una vez cerrado, modificaríamos el segundo.

¡¡Importante, para cerrar varios Issues a la vez habrá que separar cada referencia Issue por comas y volver a poner la palabra clave!! Ejemplo: “Close #34, Close #37”. Si utilizamos puntos como separación, sólo cerrará el primero.

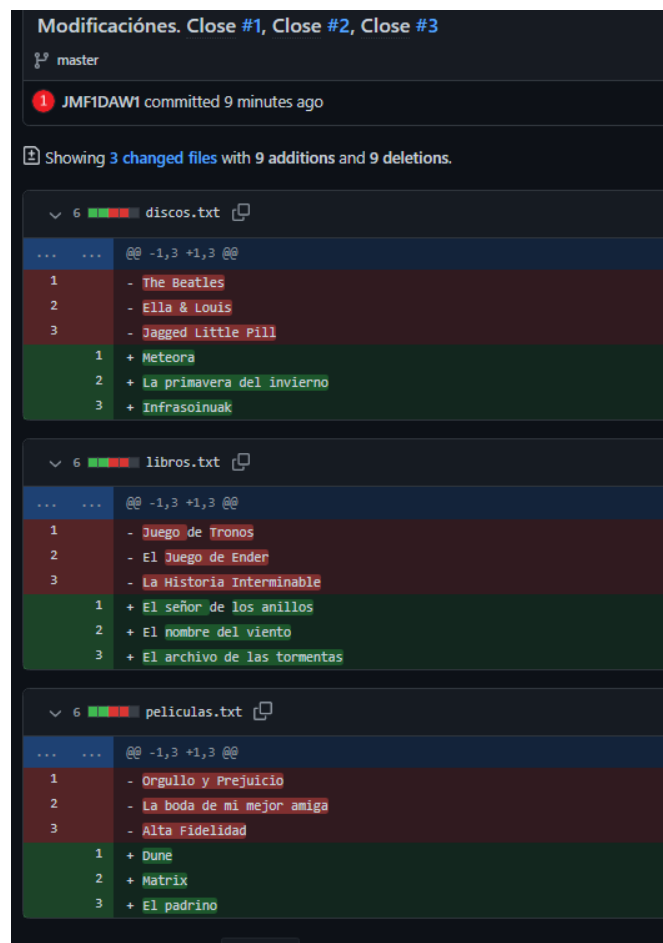
```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git commit -m "Modificaciones. Close #1, Close #2, Close #3"
[master fb14523] Modificaciones. Close #1, Close #2, Close #3
3 files changed, 9 insertions(+), 9 deletions(-)

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 558 bytes | 558.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/JMF1DAW1/sesion3-practica.git
d49efb4..fb14523 master -> master
```

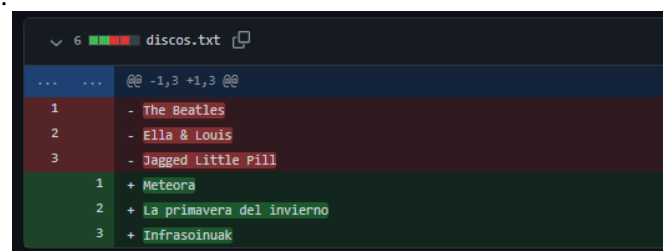
Una vez hecho el git push, actualizamos y vemos como efectivamente se ha modificado el estado de los issues de open a closed.



Si seleccionamos cualquiera de ellos, nos dirá que ha sido cerrado en y una referencia del commit. Ahí nos mostrará el commit y sus respectivos cambios.



En caso de realizar la actividad de 1 en 1, tendríamos cada Issue cerrado con la modificación del archivo concreto. El commit sería haciendo referencia a la modificación de Discos y solo usaríamos close #3.



5.e) Tal y como mostramos más arriba, los comandos que hemos utilizado han sido **git add .**, **git commit -m “Nombre y apellidos”**, y por último, **git push** para subir los cambios a GitHub.

```
Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git add .

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git commit -m "Nombre y apellidos"
[master d49efb4] Nombre y apellidos
1 file changed, 2 insertions(+)

Lau@Lau2019 MINGW64 /e/Laudrup/DAW/2º DAW/Desarrollo Entorno-Cliente/Curso-github/sesion3-practica (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/JMF1DAW1/sesion3-practica.git
 a692f8f..d49efb4 master -> master
```

5.f) Para finalizar la práctica, creamos una carpeta llamada **capturas** y añadimos este documento dentro para subirlo a GitHub.

5.g) Por último, hacemos una **Pull Request** indicando en el mensaje que hemos completado la práctica.