

## Práctica 1 - Uso de Git y GitHub

### Fundamentos de Git I

1. Instala Git en tu sistema operativo. Adjunta una captura de pantalla en la que aparezca el resultado de la ejecución del comando `git --version`.

```
> git --version  
git version 2.42.1
```

2. Realiza la configuración de Git según lo indicado en el tema (nombre, correo electrónico y editor de preferencia). Adjunta una captura de pantalla con el resultado de la ejecución de los comandos de configuración.

```
credential.helper=osxkeychain  
user.name=Alejandro Ramon  
user.email=aleramgue@alu.edu.gva.es  
core.editor=code --wait  
credential.helper=osxkeychain  
(END)
```

3. Crea una carpeta denominada S1R1. Realiza las siguientes acciones en ella:

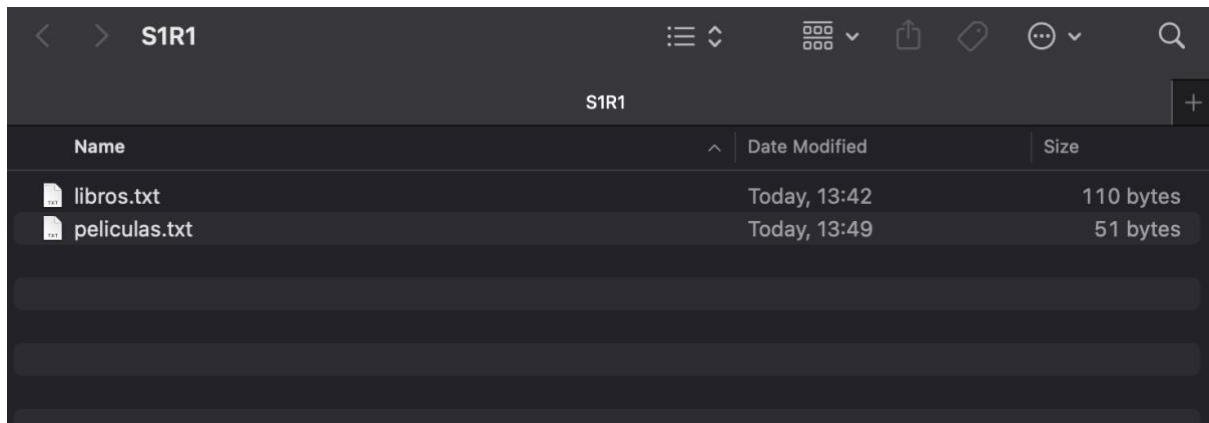
- Crea un repositorio Git.
- Crea un fichero denominado `libros.txt`. Añade tres títulos de libros cada uno en una línea distinta.
- Haz un primer commit.
- Añade dos libros al archivo `libros.txt`.
- Haz un segundo commit.
- Crea un fichero denominado `peliculas.txt`. Añade tres títulos de películas a dicho archivo.
- Haz una captura de pantalla del comando `git status`.

```
> git status  
On branch master  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    peliculas.txt
```

- Crea un fichero denominado `comidas.txt`. Añade tres nombres de comidas a dicho archivo.
- Haz un tercer commit que incluya los archivos `peliculas.txt` y `comidas.txt`.
- Elimina el archivo `comidas.txt` desde el navegador de archivos.

k) Añade dos películas más al archivo peliculas.txt.

l) Haz una captura de pantalla que muestre los cambios en el directorio de trabajo.



m) Añade los cambios al área de preparación.

n) Haz una captura de pantalla del comando git status. Debe indicar que se ha borrado el archivo comidas.txt y que se ha modificado el archivo peliculas.txt.

```

> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:      comidas.txt
        modified:     peliculas.txt
  
```

ñ) Haz un cuarto commit.

o) Crea un archivo denominado datos.bak. Añade tres títulos de libros a dicho archivo.

¡IMPORTANTE! No añadas el archivo al área de preparación ni hagas ningún commit.

p) Crea una subcarpeta denominada output. Crea un archivo denominado salida.txt en su interior. Escribe tu nombre y apellidos en dicho archivo. ¡IMPORTANTE! No añadas los archivos al área de preparación ni hagas ningún commit.

q) Haz una captura de pantalla del comando git status. Deben aparecer los archivos datos.bak y output/salida.txt como archivos nuevos (color rojo).

```

> 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)
  
```

r) Crea un archivo .gitignore para que los ficheros con extensión .bak y el contenido de la carpeta output/ no se incluyan en el repositorio.

s) Haz una nueva captura de pantalla del comando `git status`. Ahora no deben aparecer los archivos `datos.bak` y `output/salida.txt` como archivos nuevos, sino que en su lugar debe aparecer únicamente el archivo `.gitignore`.

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

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

t) Haz un último commit para incluir el archivo `.gitignore` en el repositorio.

u) Haz una captura de pantalla que muestre el histórico de cambios del repositorio.

```
* commit fb34397c34225b8f9be778f365756bcb603f1f4 (HEAD -> master)
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 14:05:49 2023 +0100
|
|     Añadimos gitignore
|
* commit 14f49cb1bd700aba7b08c292bb5fd88c7ab1f9fa
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 13:53:12 2023 +0100
|
|     Añado películas borro comidas
|
* commit b5dfddd6744b92894814bbd5b5ea1345f77957ce
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 13:48:09 2023 +0100
|
|     Películas y comidas
|
* commit 8e61a5353fd17172cf212df312ef46266e8a9e3b
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 13:43:20 2023 +0100
|
|     5 libros
|
:|
```

## FUNDAMENTOS DE GIT II

1. Crea una carpeta denominada S2R1. Realiza las siguientes acciones en ella:

- a) Crea un repositorio Git.
- b) Crea un fichero denominado actores.txt. Añade tres nombres de actores cada uno en una línea distinta.
- c) Haz un primer commit.
- d) Crea una rama denominada test.
- e) Cambia a la rama test
- f) En la rama test crea un fichero denominado actrices.txt. Añade tres nombres de actrices y realiza un commit en dicha rama.
- g) Haz una captura de pantalla del resultado del comando git log --graph --all.

```
* commit fb8f94876fd99818766068488bf551167c9647a6 (HEAD -> test)
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 14:28:32 2023 +0100
|
|     Commit en rama test
|
* commit 1d9e7063b267f0519101ce6fb803b7f87de3c6ee (master)
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 14:25:32 2023 +0100
|
|     Primer commit
|
(END)
```

- h) Cambia a la rama master.
- i) Incorpora los cambios de la rama test a la rama master. Haz una captura de pantalla de los comandos que has utilizado y de su resultado.

```
> git merge test
Updating 1d9e706..fb8f948
Fast-forward
 actrices.txt | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 actrices.txt
```

- j) Crea una segunda rama denominada test2. La rama test2 apunta al mismo commit que la rama master en este momento.
- k) En la rama master, añade una actriz al fichero actrices.txt y haz un commit.
- l) Cambia a la rama test2
- m) En la rama test2, añade una actriz al fichero actrices.txt y haz otro commit.

n) Haz una captura de pantalla del resultado del comando `git log --graph --all`. Debe haber dos caminos distintos: uno para la rama master y otro para la rama test2.

```
* commit cdd031acaa36d01533c195fd77bdbb25031b4d9b (HEAD -> test2)
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 14:38:58 2023 +0100
|
|     Añadimos actriz a test2
|
| * commit d95dcdad5c6755e8c66f3e98c2b8fc4ae988a962 (master)
|/ Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
|   Date:   Wed Nov 8 14:35:43 2023 +0100
|
|       Añadimos una actriz
|
| * commit fb8f94876fd99818766068488bf551167c9647a6 (test)
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 14:28:32 2023 +0100
|
|     Commit en rama test
|
| * commit 1d9e7063b267f0519101ce6fb803b7f87de3c6ee
| Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
| Date:   Wed Nov 8 14:25:32 2023 +0100
|
|     Primer commit
|
| (END)
```

ñ) Cambia a la rama master

o) Incorpora los cambios de la rama test2 a la rama master. ¿Se produce un conflicto? De ser así realiza una captura del comando `git status`.

```
alejandro  S2R1  master  ↓ 47
> git merge test2
Auto-merging actrizes.txt
CONFLICT (content): Merge conflict in actrizes.txt
Automatic merge failed; fix conflicts and then commit the result.
alejandro  S2R1  master  ↓ 47
> 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:   actrizes.txt

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

p) Resuelve el conflicto incorporando los dos nombres de actrices.

q) Haz una captura de pantalla del resultado del comando `git log --graph --all`. Observa que se ha creado un nuevo commit que integra los dos caminos anteriores.

```
*   commit 6b8a289ba40428978054a30578f9ec0603094b3d (HEAD -> master)
| \
|  | Merge: d95dcda cdd031a
|  | Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
|  | Date:   Wed Nov 8 14:44:06 2023 +0100
|  |
|  |     Merge branch 'test2'
|  |
| *   commit cdd031acaa36d01533c195fd77bdbb25031b4d9b (test2)
|  | Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
|  | Date:   Wed Nov 8 14:38:58 2023 +0100
|  |
|  |     Añadimos actriz a test2
|  |
| *   commit d95dcdad5c6755e8c66f3e98c2b8fc4ae988a962
| /  Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
|   Date:   Wed Nov 8 14:35:43 2023 +0100
|   |
|   |     Añadimos una actriz
|   |
| *   commit fb8f94876fd99818766068488bf551167c9647a6 (test)
|  | Author: Alejandro Ramon <aleramgue@alu.edu.gva.es>
|  | Date:   Wed Nov 8 14:28:32 2023 +0100
|  |
|  |     Commit en rama test
```

2. Crea una carpeta denominada S2R2-remoto. Inicializa un repositorio Git en su interior mediante el comando `git init --bare`. Esta carpeta se utilizará como repositorio remoto.

3. Clona el repositorio S2R2-remoto en una carpeta denominada S2R2. Adjunta captura de pantalla del resultado del comando de clonado. A continuación realiza las siguientes acciones en el repositorio S2R2:

```
> git clone S2R2-remoto S2R2
Cloning into 'S2R2'...
warning: You appear to have cloned an empty repository.
done.
```

a) Crea un archivo denominado `directores.txt`. Añade el nombre de tres directores de cine.

b) Haz un commit.

c) Realiza un push al repositorio remoto. Adjunta captura de pantalla del resultado.

```

) git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 273 bytes | 273.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To /Users/alejandro/Desktop/S2R2-remoto
 * [new branch]      master -> master

```

d) Crea una rama denominada version1.

e) Cambia a la rama version1.

f) En la rama version1 añade el nombre de dos directores de cine más al archivo directores.txt y haz un commit de los cambios.

g) Realiza un push de la rama al repositorio remoto de manera que quede asociada a la rama remota del mismo nombre. Adjunta captura de pantalla del resultado.

```

) git push --set-upstream 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), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To /Users/alejandro/Desktop/S2R2-remoto
 * [new branch]      version1 -> version1
branch 'version1' set up to track 'origin/version1'.

```

4. Clona el repositorio S2R2-remoto en una segunda carpeta denominada S2R3. Realiza las siguientes acciones sobre ella:

a) Muestra en la consola el contenido del fichero directores.txt y el resultado del comando git status. Debe mostrar tres directores.

b) Cambia a la rama version1. Muestra el resultado del comando. Comprueba que se crea una rama local version1 con el contenido de la rama remota origin/version1 y enlazada con ella. Al clonar el repositorio la rama no existía (solo se clona la rama principal, master), pero al cambiar a una rama que existe en el remoto se produce su creación local y enlazado con su correspondiente remota.

c) Muestra el contenido del fichero directores.txt por la pantalla. Comprueba que se muestran los 5 nombres de directores esperados. Adjunta captura de pantalla.

```
Martin Scorsese
Quentin Tarantino
Christopher Nolan
James Cameron
Tim Burton
directores.txt (END)
```

d) Cambia a la rama master.

e) Incorpora los cambios de la rama version1 a la rama master.

f) Sube la rama master actualizada al servidor. Adjunta captura de pantalla del resultado del comando.

```
alejandro S2R3 master ↓ 42
> git merge version1
Updating d7ef342..457dba5
Fast-forward
 directores.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
alejandro S2R3 master ↓ 41
> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To /Users/alejandro/Desktop/S2R2-remoto
 d7ef342..457dba5 master -> master
```

5. Vuelve de nuevo a la carpeta S2R2 y realiza las siguientes acciones:

a) Obtén los cambios que hay en el repositorio remoto sin fusionarlos en la rama local. Adjunta captura de pantalla del resultado del comando utilizado.

```
> git fetch origin
From /Users/alejandro/Desktop/S2R2-remoto
 d7ef342..457dba5 master -> origin/master
```

b) Actualiza la rama master local con el contenido de la rama master del repositorio remoto. Adjunta captura de pantalla del resultado del comando utilizado.

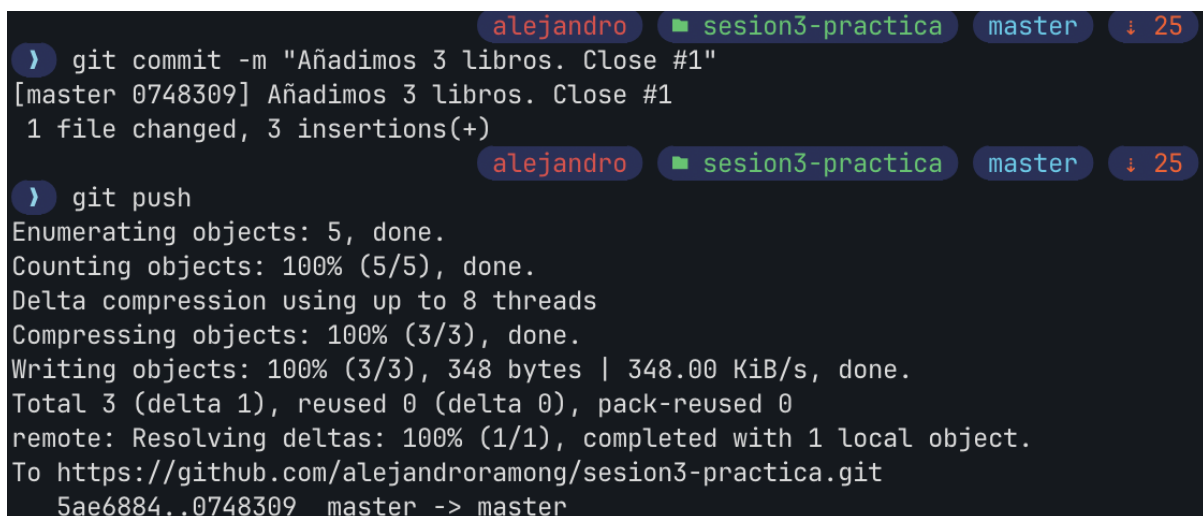
```
> git pull
Updating d7ef342..457dba5
Fast-forward
 directores.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

c) Comprueba que aparecen los 5 nombres de directores esperados.



## GitHub

1. Crea una cuenta en GitHub
2. Añade tu dirección de correo de educación
3. Solicita un descuento para uso educativo: <https://education.github.com/pack>
4. Haz un fork del repositorio localizado en la siguiente url: <https://github.com/curso-github-cefire/sesion3-practica>. A partir de este momento todas las tareas que se indican se deben realizar en tu repositorio (el que has clonado mediante el fork).
  - a) Realiza un primer commit para poner tu nombre y apellidos en el fichero README.md
  - b) Crea 3 issues con los siguientes títulos. Si no ves la pestaña de issues, actívala desde los ajustes (settings) del repositorio.
    - Añadir 3 libros
    - Añadir 3 películas
    - Añadir 3 discos
  - c) Crea una milestone denominada Tareas sesión 3-2 que contenga los 3 issues creados.
  - d) Modifica los ficheros correspondientes y realiza 3 commits para realizar cada una de las tareas que se indican en los issues. El mensaje del commit debe hacer que se cierren los issues correspondientes de manera automática.
  - e) Haz una captura de pantalla de los comandos que has utilizado para hacer los commits y subir los cambios a GitHub.



```

alejandro  ■ sesion3-practica  master  ↓ 25
➤ git commit -m "Añadimos 3 libros. Close #1"
[master 0748309] Añadimos 3 libros. Close #1
1 file changed, 3 insertions(+)

alejandro  ■ sesion3-practica  master  ↓ 25
➤ 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), 348 bytes | 348.00 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/alejandroramong/sesion3-practica.git
5ae6884..0748309  master -> master

```

```

    } git commit -m "Añadimos 3 películas. Close #2"
[master 18d3fbd] Añadimos 3 películas. Close #2
1 file changed, 3 insertions(+)

    } 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), 362 bytes | 362.00 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/alejandroramong/sesion3-practica.git
0748309..18d3fbd master -> master

    } git commit -m "Añadimos 3 discos. Close #3"
[master 0ea39ca] Añadimos 3 discos. Close #3
1 file changed, 3 insertions(+)

    } 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), 337 bytes | 337.00 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/alejandroramong/sesion3-practica.git
18d3fbd..0ea39ca master -> master

```

f) Incluye las capturas de pantalla en el repositorio dentro de la carpeta capturas. Añádelas también al repositorio de manera que queden guardadas en tu repositorio en GitHub.

g) Realiza una pull request indicando en el mensaje que has completado la tarea.