

Práctica I – Uso de Git y Github - Adrián Romero Alonso

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`.

```
windowsesp@Virtual MINGW64 ~  
$ git --version  
git version 2.42.0.windows.2
```

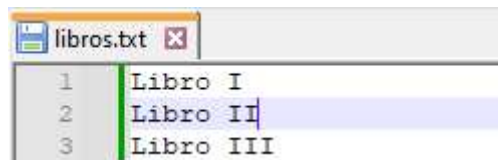
2. Realiza la configuración de Git según lo indicado en el tema (nombre, correo electrónico y editor de preferencia). Adjunta una captura de pantalla con el resultado de la ejecución de los comandos de configuración.

```
windowsesp@Virtual MINGW64 ~  
$ git config --global user.name "Adrian Romero Alonso"  
  
windowsesp@Virtual MINGW64 ~  
$ git config --global user.email adrromalo2@alu.edu.gva.es  
  
windowsesp@Virtual MINGW64 ~  
$ git config --global core.editor "C:\Program Files\Notepad++\notepad++.exe -multiInst -nosession -noPlugin"
```

3. Crea una carpeta denominada S1R1. Realiza las siguientes acciones en ella:
 - a) Crea un repositorio Git.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1  
$ git init  
Initialized empty Git repository in C:/Users/windowsesp/Desktop/Curso-Git/S1R1/.git/  
  
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
```

- b) Crea un fichero denominado `libros.txt`. Añade tres títulos de libros cada uno en una línea distinta.



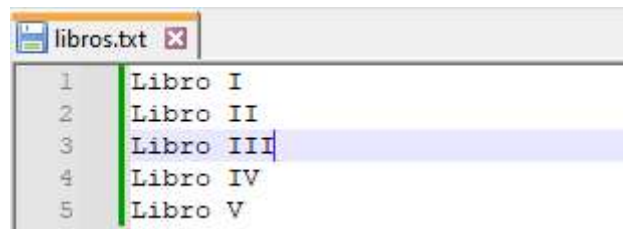
```
libros.txt  
1 Libro I  
2 Libro II  
3 Libro III
```

c) Haz un primer commit.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git add libros.txt

windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git commit -m "Primer commit"
[master (root-commit) b0b6055] Primer commit
1 file changed, 3 insertions(+)
create mode 100644 libros.txt
```

d) Añade dos libros al archivo libros.txt.



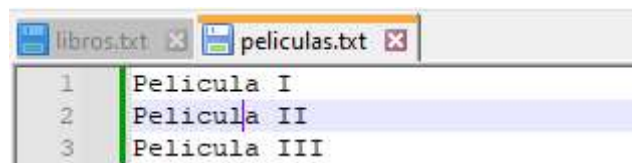
```
libros.txt
1 Libro I
2 Libro II
3 Libro III
4 Libro IV
5 Libro V
```

e) Haz un segundo commit.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git add libros.txt

windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git commit -m "Segundo commit"
[master 7c8d436] Segundo commit
1 file changed, 3 insertions(+), 1 deletion(-)
```

f) Crea un fichero denominado peliculas.txt. Añade tres títulos de películas a dicho archivo.



```
libros.txt  peliculas.txt
1 Pelicula I
2 Pelicula II
3 Pelicula III
```

g) Haz una captura de pantalla del comando git status.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ 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)

windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
```

h) Crea un fichero denominado comidas.txt. Añade tres nombres de comidas a dicho archivo.



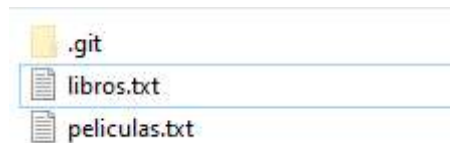
- i) Haz un tercer commit que incluya los archivos películas.txt y comidas.txt.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git add películas.txt

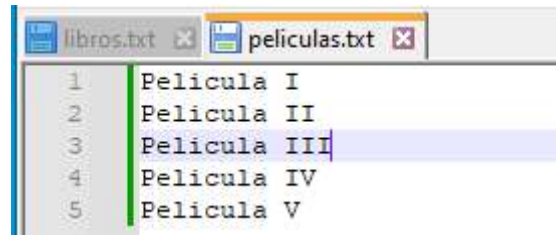
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git add comidas.txt

windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git commit -m "Tercer commit"
[master 2f76551] Tercer commit
2 files changed, 6 insertions(+)
create mode 100644 comidas.txt
create mode 100644 películas.txt
```

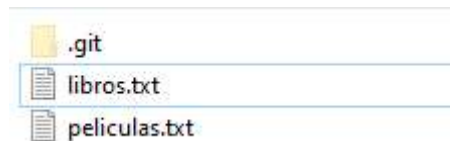
- j) Elimina el archivo comidas.txt desde el navegador de archivos.



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



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



- m) 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.

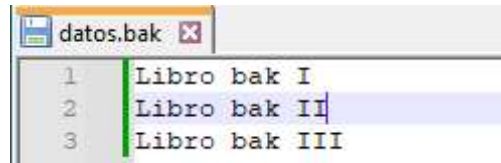
```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    comidas.txt
        modified:   peliculas.txt

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

- n) Haz un cuarto commit.

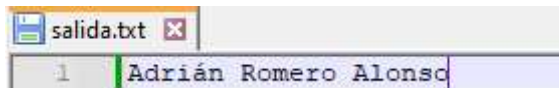
```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git commit -m "Cuarto commit"
[master 9f163be] Cuarto commit
2 files changed, 3 insertions(+), 4 deletions(-)
delete mode 100644 comidas.txt
```

- 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



A screenshot of a text editor window titled 'datos.bak'. The editor contains three lines of text, each preceded by a line number (1, 2, 3). The text is 'Libro bak I', 'Libro bak II', and 'Libro bak III'.

- 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.



A screenshot of a text editor window titled 'salida.txt'. The editor contains one line of text, preceded by a line number (1). The text is 'Adrián Romero Alonso'.

- 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).

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ 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.



```
.gitignore
1 *.bak
2 output/
```

- 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.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ 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.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git commit -m "Quinto commit"
[master 3bb731c] Quinto commit
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
```

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

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S1R1 (master)
$ git log
commit 3bb731c0f290f5be5fe4a28410e6cb5d08664b4c (HEAD -> master)
Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
Date: Tue Sep 26 16:22:48 2023 +0200

    Quinto commit

commit 9f163be6290f997ce2b69d19f05c63630e88b41d
Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
Date: Tue Sep 26 16:08:07 2023 +0200

    Cuarto commit

commit 2f765511d2c50f9ca9188ddf2a79847b7684e658
Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
Date: Tue Sep 26 15:57:46 2023 +0200

    Tercer commit

commit 7c8d436e29b0468c4585aa9fa5f6b59f9678eff5
Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
Date: Tue Sep 26 15:50:13 2023 +0200

    Segundo commit

commit b0b60552677451315b5e28b80b8f4cad5f9cb4c4
Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
Date: Tue Sep 26 15:45:17 2023 +0200

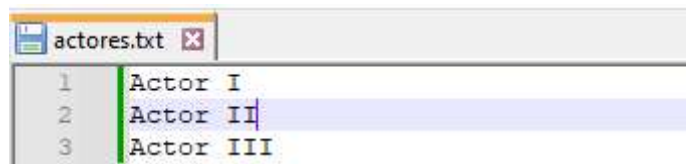
    Primer commit
```

Fundamentos de Git II

1. Crea una carpeta denominada S2R1. Realiza las siguientes acciones en ella:
 - a) Crea un repositorio Git.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1
$ git init
Initialized empty Git repository in C:/Users/windowsesp/Desktop/Curso-Git/S2R1/.git/
```

- b) Crea un fichero denominado actores.txt. Añade tres nombres de actores cada uno en una línea distinta.



```
actores.txt
1 Actor I
2 Actor II
3 Actor III
```


c) Haz un primer commit.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git commit -m "Primer commit"
[master (root-commit) 59a5884] Primer commit
1 file changed, 3 insertions(+)
create mode 100644 actores.txt
```

d) Crea una rama denominada test.

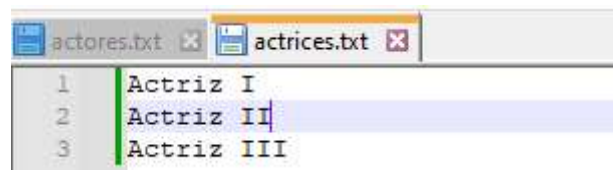
```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git branch test

windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git branch
* master
  test
```

e) Cambia a la rama test

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git checkout test
Switched to branch 'test'
```

f) En la rama test crea un fichero denominado actrices.txt. Añade tres nombres de actrices y realiza un commit en dicha rama.



```
actores.txt x actrices.txt x
1 Actriz I
2 Actriz II
3 Actriz III
```

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (test)
$ git commit -m "Segundo commit"
[test 483a3b5] Segundo commit
1 file changed, 3 insertions(+)
create mode 100644 actrices.txt
```

g) Haz una captura de pantalla del resultado del comando git log --graph -all.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (test)
$ git log --graph --all
* commit 483a3b5a8a7e92cce6defb5ded8e718dc9973ae1 (HEAD -> test)
  Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  Date:   Wed Sep 27 09:20:56 2023 +0200

      Segundo commit

* commit 59a58847b5ab51976f2d2aeb87674d28112bd546 (master)
  Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  Date:   Wed Sep 27 08:58:08 2023 +0200

      Primer commit
```

h) Cambia a la rama master.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (test)
$ git checkout master
Switched to branch '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.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git merge test
Updating 59a5884..483a3b5
Fast-forward
 actrices.txt | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 actrices.txt

windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git log
commit 483a3b5a8a7e92cce6defb5ded8e718dc9973ae1 (HEAD -> master, test)
Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
Date:   Wed Sep 27 09:20:56 2023 +0200

    Segundo commit

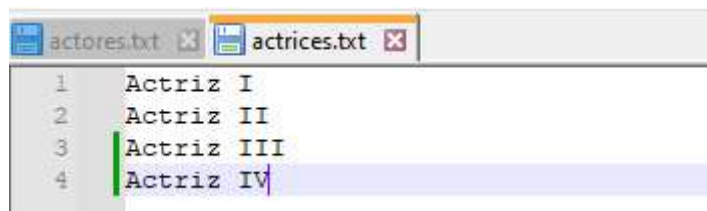
commit 59a58847b5ab51976f2d2aeb87674d28112bd546
Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
Date:   Wed Sep 27 08:58:08 2023 +0200

    Primer commit
```

j) Crea una segunda rama denominada test2. La rama test2 apunta al mismo commit que la rama master en este momento.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git branch test2
```

k) En la rama master, añade una actriz al fichero actrices.txt y haz un commit.



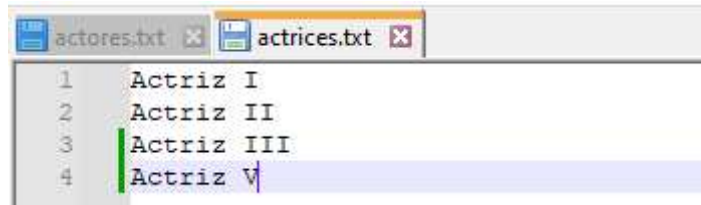
```
actrices.txt
1   Actriz I
2   Actriz II
3   Actriz III
4   Actriz IV
```

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git commit -m "Tercer commit"
[master 22b2726] Tercer commit
1 file changed, 2 insertions(+), 1 deletion(-)
```

l) Cambia a la rama test2

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git checkout test2
Switched to branch 'test2'
```

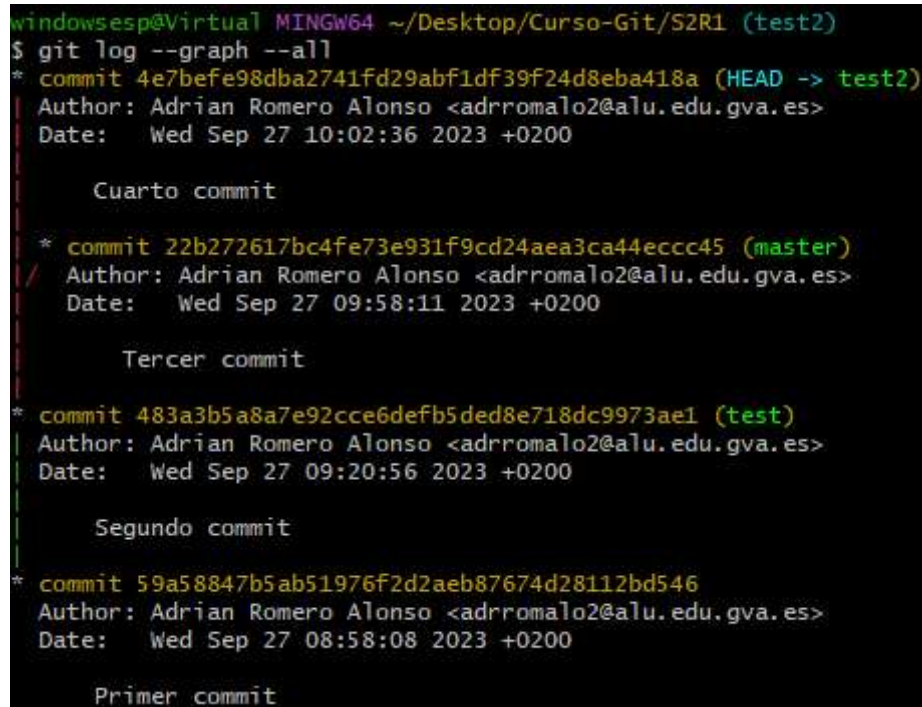
m) En la rama test2, añade una actriz al fichero actrices.txt y haz otro commit.



```
actrices.txt
1   Actriz I
2   Actriz II
3   Actriz III
4   Actriz V
```

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (test2)
$ git commit -m "Cuarto commit"
[test2 4e7befe] Cuarto commit
1 file changed, 2 insertions(+), 1 deletion(-)
```

- 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`.



```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (test2)
$ git log --graph --all
* commit 4e7befe98dba2741fd29abf1df39f24d8eba418a (HEAD -> test2)
  Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  Date:   Wed Sep 27 10:02:36 2023 +0200

    Cuarto commit

* commit 22b272617bc4fe73e931f9cd24aea3ca44eccc45 (master)
  Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  Date:   Wed Sep 27 09:58:11 2023 +0200

    Tercer commit

* commit 483a3b5a8a7e92cce6defb5ded8e718dc9973ae1 (test)
  Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  Date:   Wed Sep 27 09:20:56 2023 +0200

    Segundo commit

* commit 59a58847b5ab51976f2d2aeb87674d28112bd546
  Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  Date:   Wed Sep 27 08:58:08 2023 +0200

    Primer commit
```

- o) Cambia a la rama `master`

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (test2)
$ git checkout master
Switched to branch 'master'
```

- p) 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.

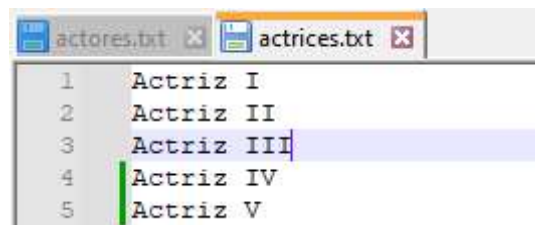
```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git merge test2
Auto-merging actrices.txt
CONFLICT (content): Merge conflict in actrices.txt
Automatic merge failed; fix conflicts and then commit the result.

windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master|MERGING)
$ 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")
```

- q) Resuelve el conflicto incorporando los dos nombres de actrices.



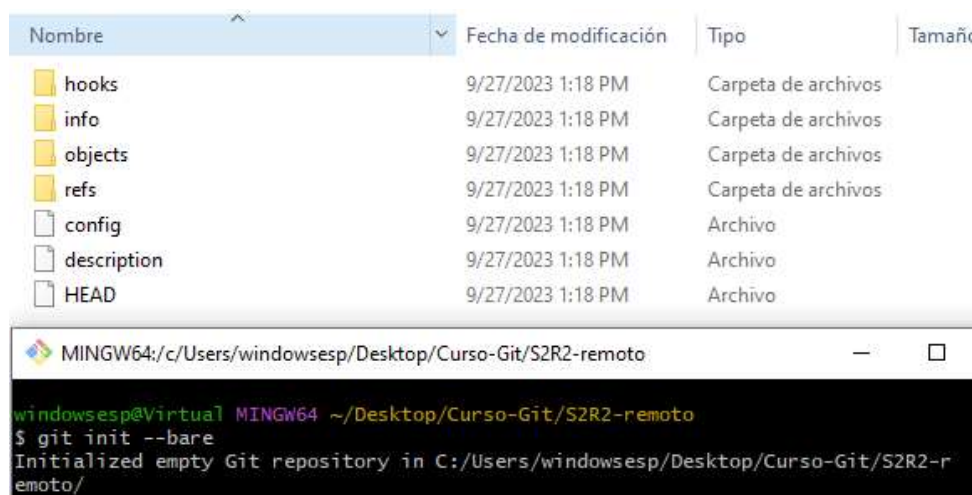
```
actrices.txt
1  Actriz I
2  Actriz II
3  Actriz III
4  Actriz IV
5  Actriz V
```

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master|MERGING)
$ git commit -m "Quinto commit"
[master 0293765] Quinto commit
```

- r) 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.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R1 (master)
$ git log --graph --all
*   commit 029376510c860fa9230ffffd62fb418956bef5d81 (HEAD -> master)
  |   Merge: 22b2726 4e7befe
  |   Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  |   Date:   Wed Sep 27 10:10:17 2023 +0200
  |
  |   Quinto commit
  |
  | *   commit 4e7befe98dba2741fd29abf1df39f24d8eba418a (test2)
  |   |   Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  |   |   Date:   Wed Sep 27 10:02:36 2023 +0200
  |   |
  |   |   Cuarto commit
  |   |
  |   | *   commit 22b272617bc4fe73e931f9cd24aea3ca44eccc45
  |   |   |   Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  |   |   |   Date:   Wed Sep 27 09:58:11 2023 +0200
  |   |   |
  |   |   |   Tercer commit
  |   |   |
  |   |   | *   commit 483a3b5a8a7e92cce6defb5ded8e718dc9973ae1 (test)
  |   |   |   |   Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  |   |   |   |   Date:   Wed Sep 27 09:20:56 2023 +0200
  |   |   |   |
  |   |   |   |   Segundo commit
  |   |   |   |
  |   |   |   | *   commit 59a58847b5ab51976f2d2aeb87674d28112bd546
  |   |   |   |   |   Author: Adrian Romero Alonso <adrromalo2@alu.edu.gva.es>
  |   |   |   |   |   Date:   Wed Sep 27 08:58:08 2023 +0200
  |   |   |   |   |
  |   |   |   |   |   Primer commit
```

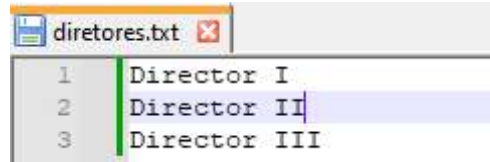
2. Crea una carpeta denominada S2R2-remoto. Inicializa un repositorio Git en su interior mediante el comando `git init --bare`. Esta carpeta se utilizará como repositorio remoto.



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:

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git
$ 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.



```
directores.txt
1 Director I
2 Director II
3 Director III
```

- b) Haz un commit.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R2 (master)
$ git commit -m "Primer commit"
[master (root-commit) c5fe4e1] Primer commit
1 file changed, 3 insertions(+)
create mode 100644 directores.txt
```

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

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R2 (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 245 bytes | 245.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To C:/Users/windowsesp/Desktop/Curso-Git/S2R2-remoto
* [new branch] master -> master
```

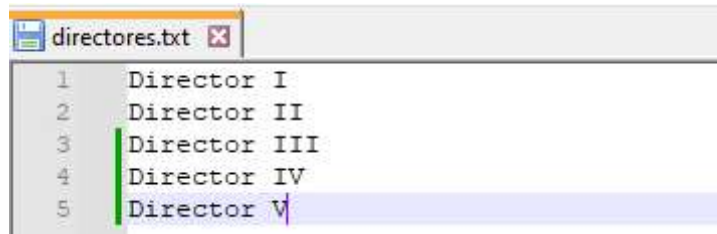
- d) Crea una rama denominada version1.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R2 (master)
$ git branch version1
```

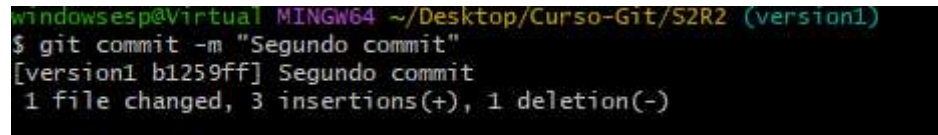
- e) Cambia a la rama version1.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R2 (master)
$ git checkout version1
Switched to branch '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.

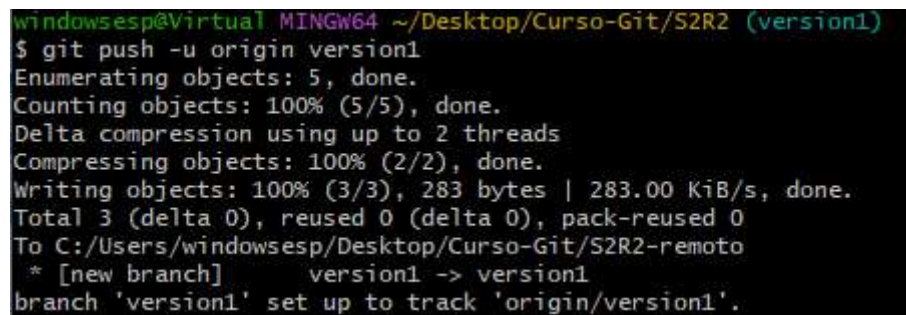


```
directores.txt
1 Director I
2 Director II
3 Director III
4 Director IV
5 Director V
```



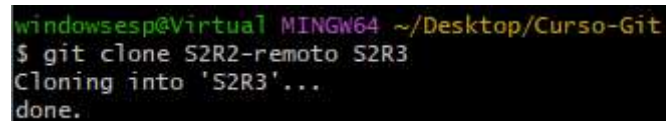
```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R2 (version1)
$ git commit -m "Segundo commit"
[version1 b1259ff] Segundo commit
1 file changed, 3 insertions(+), 1 deletion(-)
```

- 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.



```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R2 (version1)
$ git push -u origin version1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 283 bytes | 283.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To C:/Users/windowsesp/Desktop/Curso-Git/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:



```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git
$ git clone S2R2-remoto S2R3
Cloning into 'S2R3'...
done.
```


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

```
MINGW64:/c/Users/windowsest/Desktop/Curso-Git/S2R3
Director I
Director II
Director III
~
~
~
```

```
windowsest@Virtual MINGW64 ~/Desktop/Curso-Git/S2R3 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

- 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.

```
windowsest@Virtual MINGW64 ~/Desktop/Curso-Git/S2R3 (master)
$ git checkout version1
Switched to a new branch 'version1'
branch 'version1' set up to track 'origin/version1'.

windowsest@Virtual MINGW64 ~/Desktop/Curso-Git/S2R3 (version1)
$ git status
On branch version1
Your branch is up to date with 'origin/version1'.

nothing to commit, working tree clean
```

- 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.

```
MINGW64:/c/Users/windowsest/Desktop/Curso-Git/S2R3
Director I
Director II
Director III
Director IV
Director V
~
~
~
```

d) Cambia a la rama master.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R3 (version1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

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

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R3 (master)
$ git merge version1
Updating c5fe4e1..b1259ff
Fast-forward
 directorios.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

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

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R3 (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To C:/Users/windowsesp/Desktop/Curso-Git/S2R2-remoto
 c5fe4e1..b1259ff 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.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R2 (version1)
$ git fetch origin
From C:/Users/windowsesp/Desktop/Curso-Git/S2R2-remoto
 c5fe4e1..b1259ff 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.

```
windowsesp@Virtual MINGW64 ~/Desktop/Curso-Git/S2R2 (master)
$ git pull origin master
From C:/Users/windowsesp/Desktop/Curso-Git/S2R2-remoto
 * branch master -> FETCH_HEAD
Updating c5fe4e1..b1259ff
Fast-forward
 directorios.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

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

```
MINGW64:/c:/Users/windowsesp/Desktop/Curso-Git/S2R2
Director I
Director II
Director III
Director IV
Director V
~
~
~
```

Github

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).

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner *



ARomeroAlonso

Repository name *

Desarrollo-Entorno-Client

✓ Desarrollo-Entorno-Client is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Repositorio de prácticas

☒ Copy the `master` branch only

Contribute back to curso-github-cefire/sesion3-practica by adding your own branch. [Learn more.](#)

You are creating a fork in your personal account.

Create fork

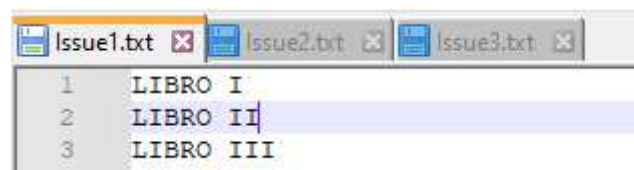
a) Realiza un primer commit para poner tu nombre y apellidos en el fichero README.md

```
1 # Uso de GitHub en la práctica docente
2 ## Práctica 3.2
3 Repositorio de prácticas para la sesión 3 del curso
4
5 Incluye a continuación tu nombre y apellidos: Adrián Romero Alonso
```

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)



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.

```

windowsest@Virtual MINGW64 ~/Desktop/Curso-Git/Desarrollo-Web-Entorno-Cliente (main)
$ git commit -m "Libros. Close#325"
[main (root-commit) ee6c2da] Libros. Close#325
3 files changed, 9 insertions(+)
create mode 100644 Issue1.txt

windowsest@Virtual MINGW64 ~/Desktop/Curso-Git/Desarrollo-Web-Entorno-Cliente (main)
$ git commit -a -m "Películas. Close #326"
[main 4a3ec6c] Películas. Close #326
1 file changed, 3 deletions(-)
delete mode 100644 Issue3.txt

windowsest@Virtual MINGW64 ~/Desktop/Curso-Git/Desarrollo-Web-Entorno-Cliente (main)
$ git commit -m "Discos. Close #327"
[main f527762] Discos. Close #327
1 file changed, 3 insertions(+)
create mode 100644 Issue3.txt

```