

# 1. 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`.

```
C:\Users\zelin>git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
           [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
           [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

C:\Users\zelin> git --version
git version 2.47.0.windows.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.

```
C:\Users\zelin> git config --global user.name "Marian Galinsoga Hernandez"
C:\Users\zelin> git config --global user.name
Marian Galinsoga Hernandez
C:\Users\zelin> git config --global user.email margalher@gmail.com
C:\Users\zelin> git config --global user.email
margalher@gmail.com
C:\Users\zelin> git config --global core.editor notepad
C:\Users\zelin> git config --global core.editor
notepad
```

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

```
C:\Users\zelin>mkdir S1R1
```

- a) Crea un repositorio Git.

```
C:\Users\zelin>cd ./S1R1
C:\Users\zelin\S1R1>git init
Initialized empty Git repository in C:/Users/zelin/S1R1/.git/
C:\Users\zelin\S1R1>
```

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

```
PS C:\Users\zelin\S1R1> # Crear el archivo y añadir títulos de libros
PS C:\Users\zelin\S1R1> Set-Content -Path "libros.txt" -Value "1. Cien años de soledad"
PS C:\Users\zelin\S1R1> Add-Content -Path "libros.txt" -Value "2. El señor de los anillos"
PS C:\Users\zelin\S1R1> Add-Content -Path "libros.txt" -Value "3. Don Quijote de la Mancha"
PS C:\Users\zelin\S1R1>
```

- c) Haz un primer *commit*.

```

PS C:\Users\zelin\S1R1> git add .
PS C:\Users\zelin\S1R1> git commit -m "Añadimos libros .txt"
[master (root-commit) b5c6ba0] Añadimos libros .txt
 1 file changed, 3 insertions(+)
 create mode 100644 libros.txt
PS C:\Users\zelin\S1R1> git log --graph --all
* commit b5c6ba04c5ce4ed8bd9a7807e98ab25f49b922c3 (HEAD -> master)
   Author: Marian Galinsoga Hernández <margalher4@gmail.com>
   Date:   Sun Nov 17 19:50:56 2024 +0100

   Añadimos libros .txt

```

d) Añade dos libros al archivo `libros.txt`.

```

PS C:\Users\zelin\S1R1> Add-Content -Path "libros.txt" -Value "4. La grieta del silencio"
PS C:\Users\zelin\S1R1> Add-Content -Path "libros.txt" -Value "5. Harry Potter y la piedra filosofal"

```

e) Haz un segundo *commit*.

```

PS C:\Users\zelin\S1R1> git add .
PS C:\Users\zelin\S1R1> git commit -m "Añadimos otros 2 libritos"
[master 804c08f] Añadimos otros 2 libritos
 1 file changed, 2 insertions(+)
PS C:\Users\zelin\S1R1> git log --graph --all
* commit 804c08f51ab00966d036d90510027837c35a95ef (HEAD -> master)
   Author: Marian Galinsoga Hernández <margalher4@gmail.com>
   Date:   Sun Nov 17 19:55:45 2024 +0100

   Añadimos otros 2 libritos

* commit b5c6ba04c5ce4ed8bd9a7807e98ab25f49b922c3
   Author: Marian Galinsoga Hernández <margalher4@gmail.com>
   Date:   Sun Nov 17 19:50:56 2024 +0100

   Añadimos libros .txt

```

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

```

PS C:\Users\zelin\S1R1> Set-Content -Path "peliculas.txt" -Value "1. El castillo ambulante"
PS C:\Users\zelin\S1R1> Add-Content -Path "peliculas.txt" -Value "2. El viaje de Chihiro"
PS C:\Users\zelin\S1R1> Add-Content -Path "peliculas.txt" -Value "3. Mi vecino Totoro"

```

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

```

PS C:\Users\zelin\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)
PS C:\Users\zelin\S1R1> git add .
PS C:\Users\zelin\S1R1> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   peliculas.txt

```

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

```

PS C:\Users\zelin\S1R1> Set-Content -Path "comidas.txt" -Value "1. Tortilla de patata"
PS C:\Users\zelin\S1R1> Add-Content -Path "comidas.txt" -Value "2. Hamburguesa"
PS C:\Users\zelin\S1R1> Add-Content -Path "comidas.txt" -Value "3. Arroz a la cubana"

```

- i) Haz un tercer *commit* que incluya los archivos `peliculas.txt` y `comidas.txt`.

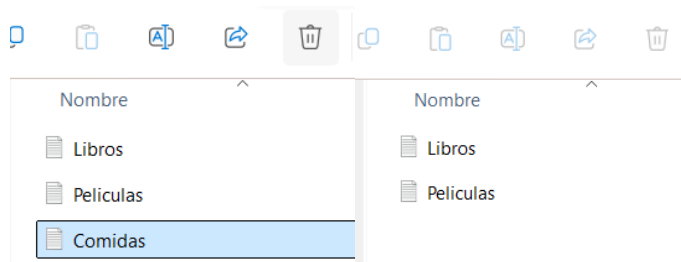
```

PS C:\Users\zelin\S1R1> git add "peliculas.txt" "comidas.txt"
PS C:\Users\zelin\S1R1> git commit -m "Añadimos comidas.txt y peliculas.txt"
[master 0c11058] Añadimos comidas.txt y peliculas.txt
 2 files changed, 6 insertions(+)
 create mode 100644 comidas.txt
 create mode 100644 peliculas.txt
PS C:\Users\zelin\S1R1> git log --graph --all
* commit 0c11058b2c6fab6c6c5b89d7ba6b3381fa644ff5e (HEAD -> master)
   Author: Marian Galinsoga Hernández <margalher4@gmail.com>
   Date:   Sun Nov 17 20:07:05 2024 +0100

   Añadimos comidas.txt y peliculas.txt

```

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



```

PS C:\Users\zelin\S1R1> 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

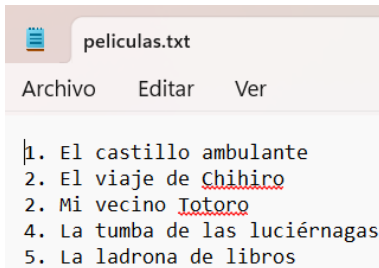
no changes added to commit (use "git add" and/or "git commit -a")

```

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

```
PS C:\Users\zelin\S1R1> Add-Content -Path "peliculas.txt" -Value "4. La tumba de las luciérnagas"
PS C:\Users\zelin\S1R1> Add-Content -Path "peliculas.txt" -Value "5. La ladrona de libros"
```

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



- m) Añade los cambios al área de preparación.

```
PS C:\Users\zelin\S1R1> git add .
```

- 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`.

```
PS C:\Users\zelin\S1R1> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:      comidas.txt
    modified:     peliculas.txt
```

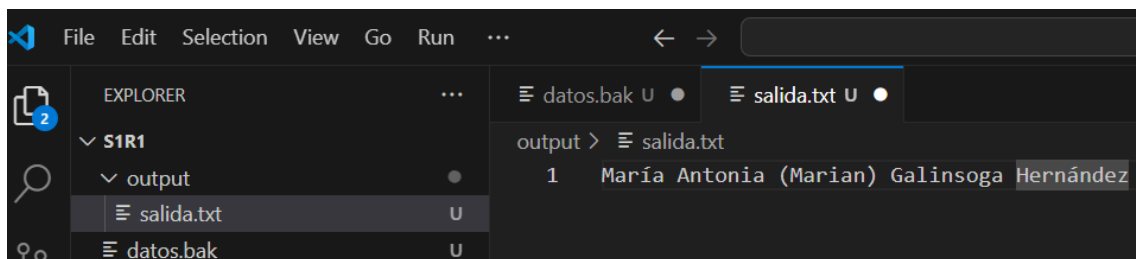
- o) Haz un cuarto *commit*.

```
S C:\Users\zelin\S1R1> git commit
int: Waiting for your editor to close the file... unix2dos: converting file C:/Users/zelin/S1R1/.git/COMMIT_EDITMSG to DOS format...
os2unix: converting file C:/Users/zelin/S1R1/.git/COMMIT_EDITMSG to Unix format...
aborting commit due to empty commit message.
S C:\Users\zelin\S1R1> git commit -m "2 cosas más hechas"
master 8952d00] 2 cosas más hechas
2 files changed, 2 insertions(+), 3 deletions(-)
delete mode 100644 comidas.txt
```

- p) 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*.

```
≡ datos.bak U ●
≡ datos.bak
1 1. Harry Potter y el prisionero de Azkaban.
2 2. Re: Zero.
3 3. Pokémon.
```

- q) 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.**



```
File Edit Selection View Go Run ...
EXPLORER
S1R1
  output
    salida.txt
    datos.bak
output > salida.txt
1 María Antonia (Marian) Galinsoga Hernández
```

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

```
PS C:\Users\zelin\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)
```

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

```
PS C:\Users\zelin\S1R1> cat .gitignore
*.bak
output/*
```

- t) 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`.

```
PS C:\Users\zelin\S1R1> 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)
```

- u) Haz un último *commit* para incluir el archivo `.gitignore` en el repositorio.


```
PS C:\Users\zelin\S1R1> git add .
PS C:\Users\zelin\S1R1> git commit -m ".gitignore"
[master bfaa03c] .gitignore
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
```


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

```
PS C:\Users\zelin\S1R1> git log --graph --all
* commit bfaa03cef2d7ae5802e928a58cbd43fac346604d (HEAD -> master)
| Author: Marian Galinsoga Hernández <margalher4@gmail.com>
| Date: Sun Nov 17 20:35:59 2024 +0100
|
|     .gitignore
|
* commit d1290d9c1c3184a24ad933dcb024b7dfe52b8f6e
| Author: Marian Galinsoga Hernández <margalher4@gmail.com>
| Date: Sun Nov 17 20:25:30 2024 +0100
|
|     .gitignore
|
* commit bfaa03cef2d7ae5802e928a58cbd43fac346604d (HEAD -> master)
| Author: Marian Galinsoga Hernández <margalher4@gmail.com>
| Date: Sun Nov 17 20:35:59 2024 +0100
|
|     .gitignore
|
```

## 2. Fundamentos de Git II

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

 S1R1

 S2R1

- a) Crea un repositorio Git.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre> cd s2r1
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git init
Initialized empty Git repository in E:/Mis cosas/Ciberseguridad/Posada de Producción Segura/Primer Trimestre/S2R1/.git/
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> cat actores.txt
1. Keanu Reeves
2. Johnny Depp
3. Daniel Radcliffe
```

- c) Haz un primer *commit*.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git commit -m "actores.txt"
[master (root-commit) cb47e7e] actores.txt
1 file changed, 3 insertions(+)
create mode 100644 actores.txt
```

- d) Crea una rama denominada `test`.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git branch
* master
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git branch test
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git branch
* master
test
```

- e) Cambia a la rama `test`.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git checkout test
Switched to branch 'test'
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git branch
master
* test
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> Set-Content -Path "actrices.txt" -Value "1. Penélope Cruz"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> Add-Content -Path "actrices.txt" -Value "2. Jennifer Aniston"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> Add-Content -Path "actrices.txt" -Value "3. Scarlett Johansson"

PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git commit -m "actrices.txt"
[test 911376e] actrices.txt
1 file changed, 3 insertions(+)
create mode 100644 actrices.txt
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git log --graph --all
* commit 911376e30186fe5038b04ec97063ea8971474b65 (HEAD -> test)
| Author: Marian Galinsoga Hernández <margalher4@gmail.com>
| Date: Sun Nov 17 20:56:25 2024 +0100
|
|     actrices.txt
|
* commit cb47e7ee6ac5300187e7d5f12c040ecc3564c0fc (master)
| Author: Marian Galinsoga Hernández <margalher4@gmail.com>
| Date: Sun Nov 17 20:48:41 2024 +0100
```

- h) Cambia a la rama `master`.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> 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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git merge test
Updating cb47e7e..911376e
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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git branch test2
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git checkout test2
Switched to branch 'test2'
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git status
On branch test2
nothing to commit, working tree clean
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git log --graph --all
* commit 911376e30186fe5038b04ec97063ea8971474b65 (HEAD -> test2, test, master)
| Author: Marian Galinsoga Hernández <margalher4@gmail.com>
| Date: Sun Nov 17 20:56:25 2024 +0100
|
| actrices.txt
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> Add-Content -Path "actrices.txt" -Value "4. Emma Watson"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git commit -m "Actriz +1"
[master 69119e4] Actriz +1
1 file changed, 1 insertion(+)
```

- l) Cambia a la rama `test2`.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git checkout test2
Switched to branch 'test2'
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> Add-Content -Path "actrices.txt" -Value "5. Ana de Armas"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git commit -m "Actriz +armas"
[test2 44c152a] Actriz +armas
1 file changed, 1 insertion(+)
```

- 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`.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git log --graph --all
* commit 44c152ae07c535fad4caad33d6c5a7d3b6f41f65 (HEAD -> test2)
| Author: Marian Galinsoga Hernández <margalher4@gmail.com>
| Date: Sun Nov 17 21:09:49 2024 +0100
|
| Actriz +armas
|
| * commit 69119e40d7d0a35efa10b1bbe481d3111ab5734e (master)
| / Author: Marian Galinsoga Hernández <margalher4@gmail.com>
| / Date: Sun Nov 17 21:07:58 2024 +0100
|
| Actriz +1
```



o) Cambia a la rama master.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> 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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git merge test2
Auto-merging actrices.txt
CONFLICT (content): Merge conflict in actrices.txt
Automatic merge failed; fix conflicts and then commit the result.
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> 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
```

Sí, se produce un conflicto.

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

```
4 Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
5 <<<<<< HEAD (Current Change)
6 4. Emma Watson
7 =====
8 >>>>>> test2 (Incoming Change)
9 5. Ana de Armas
```

```
≡ actores.txt  ≡ actrices.txt X
S2R1 > ≡ actrices.txt
1 1. Penelope Cruz
2 2. Jennifer Aniston
3 3. Scarlett Johansson
4 4. Emma Watson
5 5. Ana de Armas
```

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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\1. Fundamentos de Git I, Fundamentos de Git II y GitHub\s2r1> git log
--graph --all
* commit 44c152ae07c535fad4caad33d6c5a7d3b6f41f65 (test2)
 | Author: Marian Galinsoga Hernández <margalher4@gmail.com>
 | Date: Sun Nov 17 21:09:49 2024 +0100
 |
 | ...skipping...
 | * commit 44c152ae07c535fad4caad33d6c5a7d3b6f41f65 (test2)
 | | Author: Marian Galinsoga Hernández <margalher4@gmail.com>
 | | Date: Sun Nov 17 21:09:49 2024 +0100
 | |
 | | Actriz +armas
 |
 | * commit 69119e40d7d0a35efa10b1bbe481d3111ab5734e (HEAD -> master)
 | / Author: Marian Galinsoga Hernández <margalher4@gmail.com>
 | / Date: Sun Nov 17 21:07:58 2024 +0100
 | /
 | / Actriz +1
```

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r1> git status
On branch master
All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  modified:   actrices.txt

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:   actrices.txt
```

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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2-remoto> git init --bare
Initialized empty Git repository in E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\S2R2-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:

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre> 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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> Set-Content -Path "directores.txt" -Value "1. Steven Spielberg"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> Add-Content -Path "directores.txt" -Value "2. Martin Scorsese"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> Add-Content -Path "directores.txt" -Value "3. Woody Allen"
```

- b) Haz un *commit*.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> git commit -m "directores.txt"
[master (root-commit) b46113e] directores.txt
 1 file changed, 3 insertions(+)
 create mode 100644 directores.txt
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> 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), 293 bytes | 41.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2-remoto/
* [new branch]      master -> master
```

d) Crea una rama denominada `version1`.

```
[new branch]      master -> master
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> git branch version1
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2>
```

e) Cambia a la rama `version1`.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> 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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> Add-Content -Path "directores.txt" -Value "4. Tim Burton"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> Add-Content -Path "directores.txt" -Value "5. Pedro Almodóvar"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> git commit -m "directores +2"
[version1 d7af9e6] directores +2
1 file changed, 2 insertions(+)
```

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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> git push ../s2r2-remoto 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), 353 bytes | 176.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To ../s2r2-remoto
* [new branch]      version1 -> version1
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre> 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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre> cat s2r3/directores.txt
1. Steven Spielberg
2. Martin Scorsese
3. Woody Allen

PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r3> 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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r3> git checkout version1
branch 'version1' set up to track 'origin/version1'.
Switched to a new branch 'version1'
```

- 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.

```
Switched to a new branch 'version1'
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r3> cat directores.txt
1. Steven Spielberg
2. Martin Scorsese
3. Woody Allen
4. Tim Burton
5. Pedro Almodóvar
```

- d) Cambia a la rama `master`.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r3> 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`.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\1. Fundamentos de Git I, Fundamentos de Git II y GitHub\s2r3> git push
h -u origin version1
Updating b46113e..d7af9e6
Fast-forward
 directores.txt | 2 ++
 1 file changed, 2 insertions(+)
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r3> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To E:/Mis cosas/Ciberseguridad/Posada de Producción Segura/Primer Trimestre/s2r2-remoto/
 b46113e..d7af9e6 master -> master
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r3> cd ..
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre> cd s2r2
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2>
```

- 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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> git fetch
From E:/Mis cosas/Ciberseguridad/Posada de Producción Segura/Primer Trimestre/s2r2-remoto
 b46113e..d7af9e6 master -> origin/master
* [new branch]      version1 -> origin/version1
```

- 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.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> git pull origin master
From E:/Mis cosas/Ciberseguridad/Posada de Producción Segura/Primer Trimestre/s2r2-remoto
* branch          master      -> FETCH_HEAD
Already up to date.
```

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\s2r2> cat directores.txt
1. Steven Spielberg
2. Martin Scorsese
3. Woody Allen
4. Tim Burton
5. Pedro Almodóvar
```

### 3. GitHub

1. Crea una cuenta en GitHub.



Iniciar sesión en GitHub

Tu cuenta se ha creado correctamente. Inicia sesión para continuar. ×

Nombre de usuario o dirección de correo electrónico

MarianGalinsoga

Contraseña [¿Has olvidado tu contraseña?](#)

.....

Iniciar sesión

2. Añade tu dirección de correo de educación.



Panel

Su correo electrónico fue verificado.

3. Solicita un descuento para uso educativo:  
<https://education.github.com/pack>

## Paquete para estudiantes y desarrolladores de GitHub

Aprenda a enviar software como un profesional. No hay nada que sustituya a la experiencia práctica. Pero para la mayoría de los estudiantes, las herramientas del mundo real pueden resultar prohibitivas en cuanto a costos. Por eso, creamos el paquete para desarrolladores de estudiantes de GitHub con algunos de nuestros socios y amigos.

Regístrate para el paquete de desarrollador para estudiantes

[Hogar](#) / [Solicitud de beneficios](#)

## Acceda a los beneficios gratuitos de GitHub Education

Complete los campos a continuación para desbloquear herramientas y recursos para su trayectoria educativa.

Seleccione su rol en la educación \*

☐  Maestro ☒  Alumno ☐  Escuela

### Solicitud

Los campos obligatorios están marcados con un asterisco \*

¿Cómo se llama tu escuela? \*

**Nota:** Si su escuela no aparece en la lista, ingrese el nombre completo de la escuela y continúe. Se le solicitará que proporcione más información sobre su escuela en la página siguiente. **Se requiere un mínimo de dos caracteres para encontrar su escuela.**

Secondary School Mare Nostrum

[Hogar](#) / [Obtenga beneficios](#)

## ¡Gracias por enviarnos!

Asegúrate de revisar tu correo electrónico. Si no recibes noticias nuestras en una hora, deberías recibir un correo electrónico nuestro en menos de **10 días**. ¡Que tengas un día fantástico!

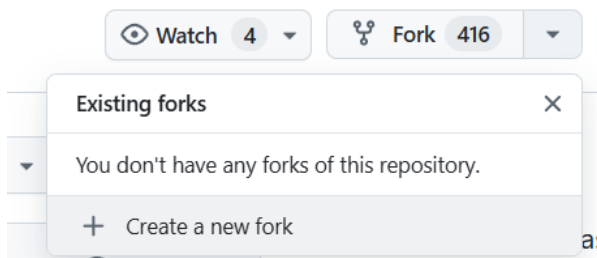
# Ya has enviado 1 solicitud

**Rechazado**

Enviado el 17 de  
noviembre de 2024

Lamentablemente, no pudimos aprobar su  
solicitud de descuento educativo para  
[@MarianGalinsoga](#).

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*).




## Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (\*).

Owner \*      Repository name \*

 MarianGalinsoga / sesion3-practica

✓ sesion3-practica is available.

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

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre> git clone https://github.com/curso-github-cefire/sesion3-practica
Cloning into 'sesion3-practica'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 6 (from 1)
Receiving objects: 100% (10/10), done.
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre> cd .\sesion3-practica\
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git commit -m "Nombre y apellido"
[master a3f828d] Nombre y apellido
1 file changed, 1 insertion(+), 1 deletion(-)
```



- 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

☐
☒ 3 Abierto
 ☒ 0 Cerrado

☐
☒ Añadir 3 discos  
 #3 abiertoahorapor MarianGalinsoga

☐
☒ Añadir 3 películas  
 #2 abierto hace 1 minutopor MarianGalinsoga

☐
☒ Añadir 3 libros  
 #1 abierto hace 1 minutopor MarianGalinsoga

- c) Crea una milestone denominada Tareas sesión 3-2 que contenga los 3 *issues* creados.

☒ 1 Open
 ☒ 0 Closed

Tareas sesión 3-2  
 No due date
  Last updated less than a minute ago

☒ 3 Open
 ☒ 0 Closed

☒ Añadir 3 discos  
 #3 opened 8 minutes ago by MarianGalinsoga
 ☒ Tareas sesión 3-2

☒ Añadir 3 películas  
 #2 opened 9 minutes ago by MarianGalinsoga
 ☒ Tareas sesión 3-2

☒ Añadir 3 libros  
 #1 opened 9 minutes ago by MarianGalinsoga
 ☒ Tareas sesión 3-2

- 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.

```

PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> Add-Content -Path "discos.txt" -Value "1. Linkin Park"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> Add-Content -Path "discos.txt" -Value "2. Rolling Stone"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> Add-Content -Path "discos.txt" -Value "3. The Beatles"
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git commit -m "Se cierra issue, discos. close #3"
[master ef84712] Se cierra issue, discos. close #3
2 files changed, 4 insertions(+), 1 deletion(-)
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git push
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 555 bytes | 277.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
  
```

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git commit -m "Se cierra issue, libros. close #1"
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git push
Everything up-to-date
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git commit -m "Se cierra issue, películas. close #2"
[master 98a1b88] Se cierra issue, películas. close #2
1 file changed, 3 insertions(+)
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git push
Enumerating objects: 5, done.
```

- e) Haz una captura de pantalla de los comandos que has utilizado para hacer los commits y subir los cambios a GitHub.

```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git push
Everything up-to-date
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git commit -m "Se cierra issue, películas. close #2"
[master 98a1b88] Se cierra issue, películas. close #2
1 file changed, 3 insertions(+)
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git push
Enumerating objects: 5, done.
```

**GIT ADD, GIT COMMIT -M “SE CIERRA ISSUE, NOMBRE. CLOSE #NUM”, GIT PUSH.**

- 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.



```
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git status
On branch master
Your branch is up to date with 'origin/master'.





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


nothing added to commit but untracked files present (use "git add" to track)
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git add .
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> git commit -m "Agregamos LA captura"
[master d7b81f8] Agregamos LA captura
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Capturas/Captura Shiny.jpg
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica> 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% (4/4), 390.97 KiB | 15.04 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/MarianGalinsoga/sesion3-practica
98a1b88..d7b81f8 master -> master
PS E:\Mis cosas\Ciberseguridad\Posada de Producción Segura\Primer Trimestre\sesion3-practica>
```

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

# TAREA COMPLETADA!!! #494


 Open MarianGalinsoga wants to merge 4 commits into `curso-github-cefire:master` from `MarianGalinsoga:master` 






 Conversation **0**  Commits **4**  Checks **0**  Files changed **5**



**MarianGalinsoga** commented now

*No description provided.*



-  **Marian Galinsoga Hernández** added 4 commits 18 minutes ago
-  Se cierra issue, discos. [close #3](#)
  -  Se cierra issue, discos. [close #1](#)
  -  Se cierra issue, películas. [close #2](#)
  -  Agregamos LA captura

## 4. Entrega de la tarea

Crea un documento con las capturas pedidas (Fundamentos de Git I, Fundamentos de Git II y GitHub) y guárdalo en formato **PDF**.

Guarda dicho documento en la carpeta `capturas` del repositorio y sube los cambios a GitHub. La entrega de la tarea se realiza a través de la `pull request` en GitHub.