

PRÁCTICA 1



Jorge Cabrera Ortiz
1 CES Puesta en Producción

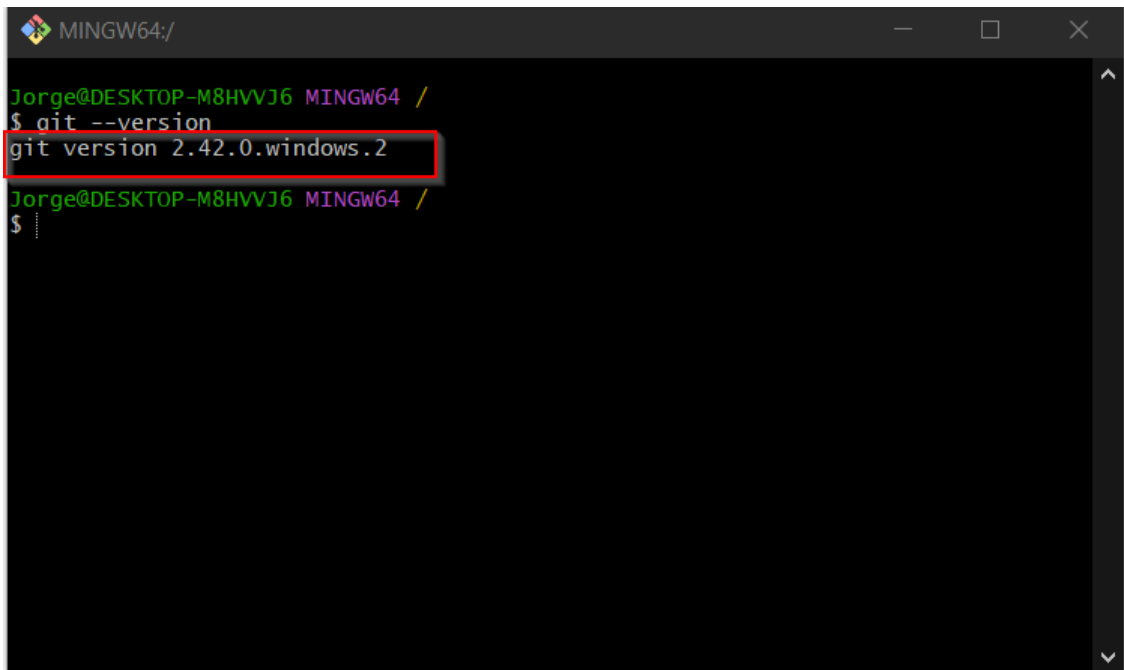
Indice

Contenido

2. Fundamentos de Git I.....	2
3. Fundamentos de Git II.....	9
4. GitHub.....	20

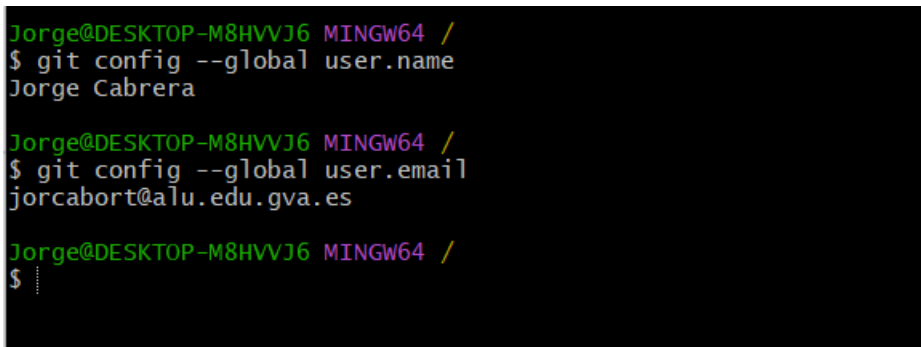
2. Fundamentos de Git I

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`.



```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /
$ git --version
git version 2.42.0.windows.2
Jorge@DESKTOP-M8HVVJ6 MINGW64 /
$
```

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.

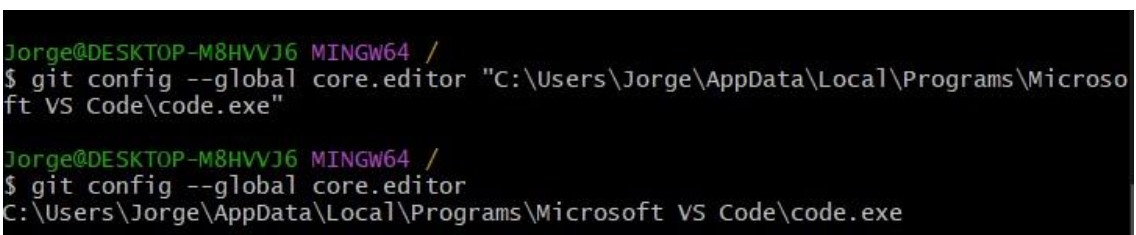


```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /
$ git config --global user.name
Jorge Cabrera

Jorge@DESKTOP-M8HVVJ6 MINGW64 /
$ git config --global user.email
jorcabort@alu.edu.gva.es

Jorge@DESKTOP-M8HVVJ6 MINGW64 /
$
```

Editor a utilizar visual studio



```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /
$ git config --global core.editor "C:\Users\Jorge\AppData\Local\Programs\Microsoft VS Code\code.exe"

Jorge@DESKTOP-M8HVVJ6 MINGW64 /
$ git config --global core.editor
C:\Users\Jorge\AppData\Local\Programs\Microsoft VS Code\code.exe
```

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

```
MINGW64:/C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
$ pwd
/C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
```

a) Crea un repositorio Git.

```
MINGW64:/C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
$ git init
Initialized empty Git repository in C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1/.git/

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$
```

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

```
libros.txt X
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S1R1 > libros.txt
1 El monje que vendio su ferrari
2 Vivir con abundancia
3 La voz de tu alma
```

c) Haz un primer commit.

```
$ git add libros.txt

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   libros.txt

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git commit -m "tres titulos"
[master (root-commit) 3c4fa09] tres titulos
1 file changed, 3 insertions(+)
create mode 100644 libros.txt
```

d) Añade dos libros al archivo libros.txt.

```
libros.txt X
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S1R1 > libros.txt
1 El monje que vendió su ferrari
2 Vivir con abundancia
3 La voz de tu alma
4 Habitos atomicos
5 El arte de la guerra
```

e) Haz un segundo commit.

```
MINGW64:/C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git add libros.txt

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git commit -m "Dos titulos mas"
[master e550698] Dos titulos mas
1 file changed, 3 insertions(+), 1 deletion(-)

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git status
On branch master
nothing to commit, working tree clean
```

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

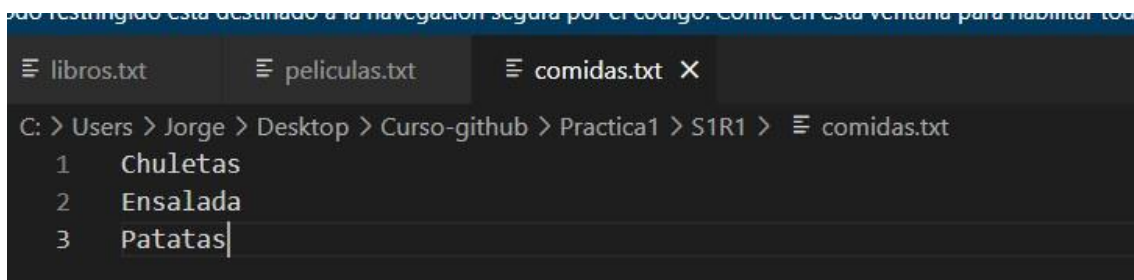
```
libros.txt peliculas.txt X
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S1R1 > peliculas.txt
1 El padrino
2 Gladiator
3 A todo gas|
```

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

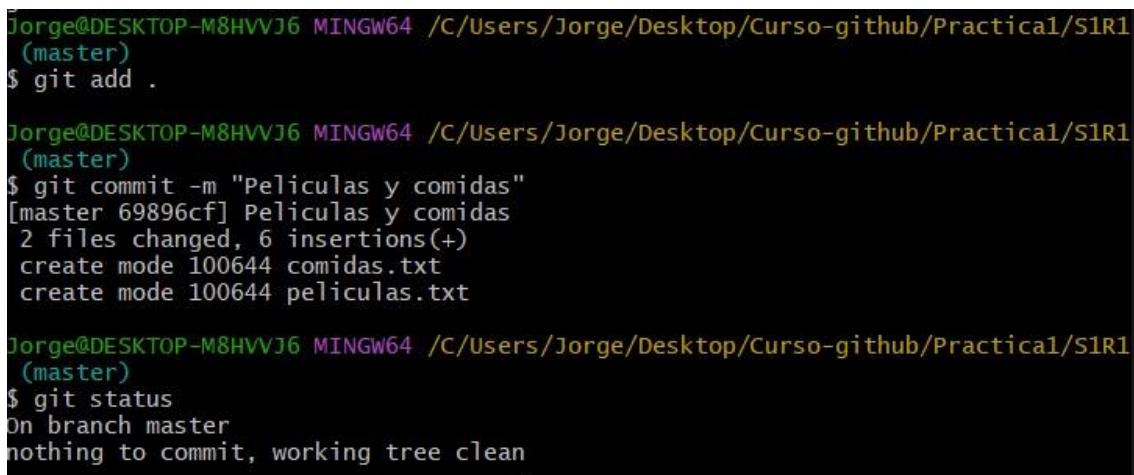
```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    peliculas.txt

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

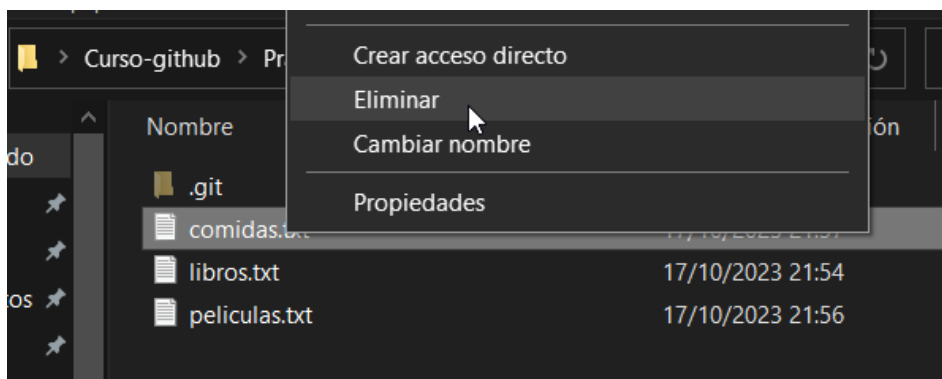
h) Crea un fichero denominado comidas.txt. Añade tres nombres de comidas a dicho archivo.



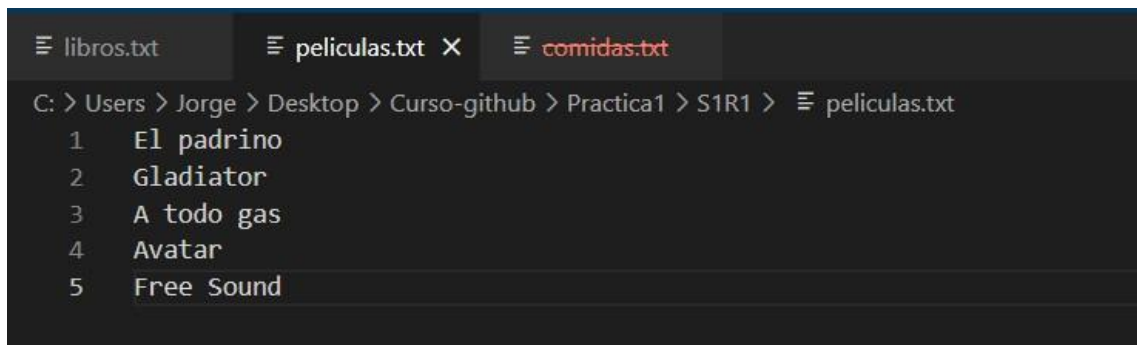
i) Haz un tercer commit que incluya los archivos películas.txt y comidas.txt.



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

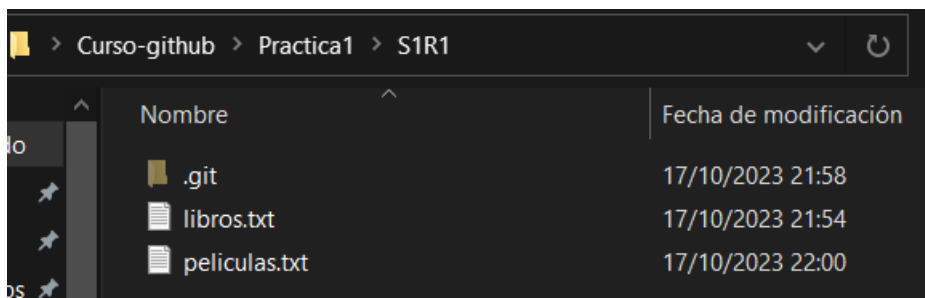


k) Añade dos películas más al archivo peliculas.txt.




```
libros.txt  peliculas.txt  comidas.txt
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S1R1 > peliculas.txt
1  El padrino
2  Gladiator
3  A todo gas
4  Avatar
5  Free Sound
```

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

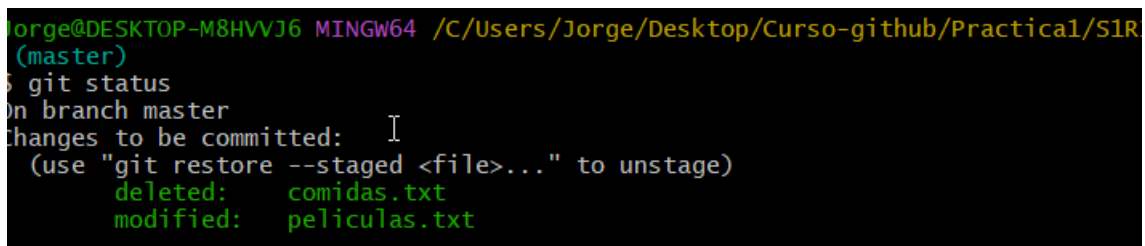


m) Añade los cambios al área de preparación.



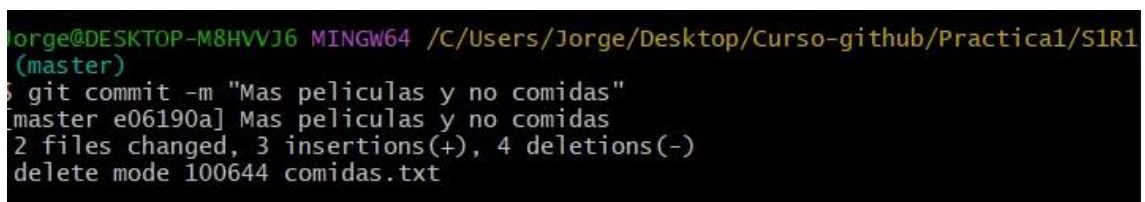
```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git add .
```

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.



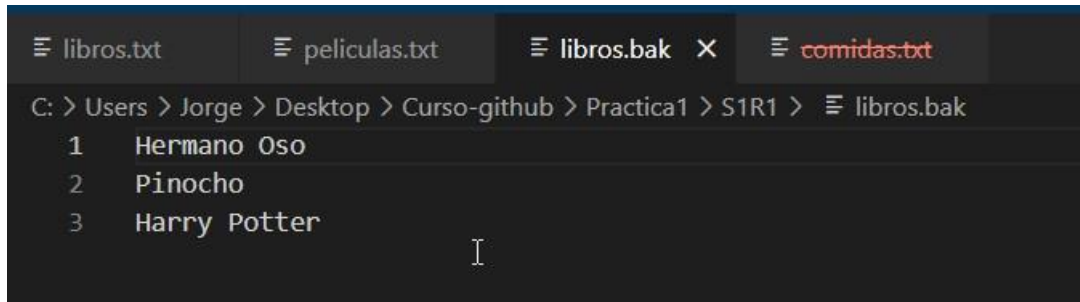
```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git status
On branch master
changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   comidas.txt
    modified:  peliculas.txt
```

ñ) Haz un cuarto commit.



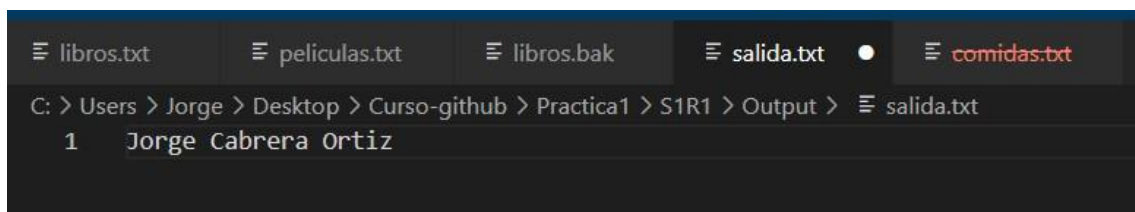
```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git commit -m "Mas peliculas y no comidas"
[master e06190a] Mas peliculas y no comidas
2 files changed, 3 insertions(+), 4 deletions(-)
delete mode 100644 comidas.txt
```


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.



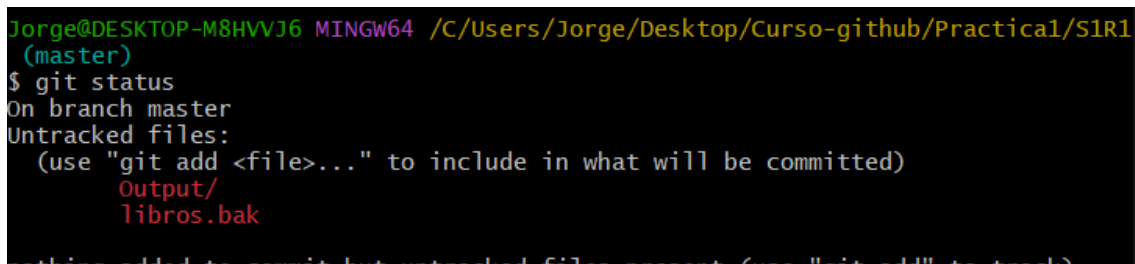
```
libros.txt  peliculas.txt  libros.bak  comidas.txt
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S1R1 > libros.bak
1 Hermano Oso
2 Pinocho
3 Harry Potter
```

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.



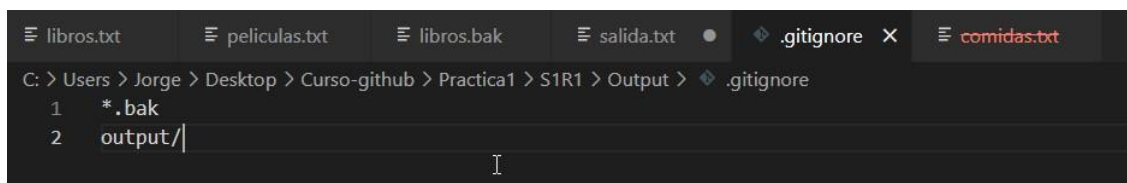
```
libros.txt  peliculas.txt  libros.bak  salida.txt  comidas.txt
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S1R1 > Output > salida.txt
1 Jorge Cabrera Ortiz
```

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).



```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    output/
    libros.bak
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.



```
libros.txt  peliculas.txt  libros.bak  salida.txt  .gitignore  comidas.txt
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S1R1 > Output > .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.


```

$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    output/
    libros.bak

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

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C/Users/Jorge/Desktop/Curso-github/Practica1/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.

```

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git add .

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C/Users/Jorge/Desktop/Curso-github/Practica1/S1R1
(master)
$ git commit -m "gitignore incluido"
[master 47f6971] gitignore incluido
1 file changed, 2 insertions(+)
create mode 100644 .gitignore

```

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

```

commit 47f69712/cddfea24d482cd8c22af1eeb6a/0d1b (HEAD -> master)
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Tue Oct 17 22:16:58 2023 +0200

    gitignore incluido

commit e06190af84cd65a89dc2d9c5b8a0811cf94074c9
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Tue Oct 17 22:03:20 2023 +0200

    Mas peliculas y no comidas

commit 69896cf01de20794d573d5f7bd4ab43e7f6a8c20
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Tue Oct 17 21:58:33 2023 +0200

    Peliculas y comidas

commit e550698d6c1138b99a103e60f69a85b8b6b098cd
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Tue Oct 17 21:54:56 2023 +0200

    Dos titulos mas

commit 3c4fa098c54e4f3627c92bdaa474ccdd9f8f152a
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Tue Oct 17 21:50:22 2023 +0200

    tres titulos

```

3. Fundamentos de Git II

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

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1
$ pwd
/c/Users/Jorge/Desktop/Curso-github/Practica1/S2R1

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1
$ |
```

a) Crea un repositorio Git.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1
$ git init
Initialized empty Git repository in C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R1/.git/
```

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

```
actores.txt X salida.txt ●
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S2R1 > actores.txt
1 Tom Cruise
2 Tom Hank
3 Samuel Jackson
```

c) Haz un primer commit.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git add actores.txt

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git commit -m "Tres actores"
[master (root-commit) ac762d8] Tres actores
1 file changed, 3 insertions(+)
create mode 100644 actores.txt
```

d) Crea una rama denominada test.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git branch test

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git branch
* master
  test
```

e) Cambia a la rama test

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git checkout test
Switched to branch 'test'

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test)
$ 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.

```
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S2R1 > actrices.txt
1 Angelina Jolie
2 Scarlett johansson
3 Penelope Cruz|
```

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test)
$ git branch
  master
* test

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test)
$ git add actrices.txt

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test)
$ git commit -m "Tres actrices"
[test 39cb716] Tres actrices
1 file changed, 3 insertions(+)
create mode 100644 actrices.txt
```

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

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test)
$ git log --graph --all
* commit 39cb716935ebda6b75381220133b91209ac6dcb4 (HEAD -> test)
  Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
  Date:   Wed Oct 18 21:01:58 2023 +0200

    Tres actrices

* commit ac762d81edf01705dd1ad276ceac3cb84b2bdca2 (master)
  Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
  Date:   Wed Oct 18 20:55:35 2023 +0200

    Tres actores
```

h) Cambia a la rama master.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test)
$ git checkout master
Switched to branch 'master'

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git branch
  master
* test
```

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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git merge test
Updating ac762d8..39cb716
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.

```
orge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
git branch test2

orge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
git branch
  master
  test
  test2
```

```
orge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
git log
commit 39cb716935ebda6b75381220133b91209ac6dcb4 (HEAD -> master, test2, test)
author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
date:   Wed Oct 18 21:01:58 2023 +0200

    Tres actrices

commit ac762d81edf01705dd1ad276ceac3cb84b2bdca2
author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
date:   Wed Oct 18 20:55:35 2023 +0200

    Tres actores
```

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

```
orge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
git status
On branch 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:   actrices.txt

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

orge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
git add actrices.txt

orge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
git commit -m "Se añade una 4 actriz"
[master b67fad3] Se añade una 4 actriz
1 file changed, 2 insertions(+), 1 deletion(-)
```

l) Cambia a la rama test2

```
orge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git checkout test2
Switched to branch 'test2'

orge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test2)
$ git status
On branch test2
nothing to commit, working tree clean
```


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

```
jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test2)
git status
On branch test2
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

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

jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test2)
git add actrices.txt

jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test2)
git commit -m "otra actriz mas"
[test2 4845a27] otra actriz mas
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.

```
jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test2)
git log --graph --all
commit 4845a271388ce32039c213cf8e3d689b1db7a30e (HEAD -> test2)
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date:   Wed Oct 18 21:19:25 2023 +0200

    otra actriz mas

* commit b67fad33c72f2df9ff5f24e667436963a1a4d30c (master)
/ Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
  Date:   Wed Oct 18 21:16:31 2023 +0200

    Se añade una 4 actriz

commit 39cb716935ebda6b75381220133b91209ac6dcb4 (test)
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date:   Wed Oct 18 21:01:58 2023 +0200

    Tres actrices

commit ac762d81edf01705dd1ad276ceac3cb84b2bdca2
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date:   Wed Oct 18 20:55:35 2023 +0200

    Tres actores
```

Se resuelve el conflicto:

```
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

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master|MERGING)
$ git add .

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master|MERGING)
$ git commit -m "resuelvo conflicto"
[master b11d8db] resuelvo conflicto

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git log -graph--log
fatal: unrecognized argument: -graph--log

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git log --graph --all
*   commit b11d8db2029865e4b3e4051ce57ea51da889e391 (HEAD -> master)
    | Merge: b67fad3 4845a27
    | Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
    | Date: Tue Nov 7 16:40:19 2023 +0100
    |
    |     resuelvo conflicto
    |
    *   commit 4845a271388ce32039c213cf8e3d689b1db7a30e (test2)
    | Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
    | Date: Wed Oct 18 21:19:25 2023 +0200
    |
    |     otra actriz mas
    |
    *   commit b67fad33c72f2df9ff5f24e667436963a1a4d30c
    | Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
    | Date: Wed Oct 18 21:16:31 2023 +0200
    |
    |     Se añade una 4 actriz
    |
    *   commit 39cb716935ebda6b75381220133b91209ac6dcb4 (test)
    | Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
    | Date: Wed Oct 18 21:01:58 2023 +0200
    |
    |     Tres actrices
    |
    *   commit ac762d81edf01705dd1ad276ceac3cb84b2bdca2
    | Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
    | Date: Wed Oct 18 20:55:35 2023 +0200
    |
    |     Tres actores
```

ñ) Cambia a la rama master

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (test2)
$ git checkout master
Switched to branch 'master'

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master)
$ git log
commit b67fad33c72f2df9ff5f24e667436963a1a4d30c (HEAD -> master)
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Wed Oct 18 21:16:31 2023 +0200

    Se añade una 4 actriz

commit 39cb716935ebda6b75381220133b91209ac6dcb4 (test)
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Wed Oct 18 21:01:58 2023 +0200

    Tres actrices

commit ac762d81edf01705dd1ad276ceac3cb84b2bdca2
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Wed Oct 18 20:55:35 2023 +0200

    Tres actores
```

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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/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.

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/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")
```

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

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master|MERGING)
$ git add .

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master|MERGING)
$ 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
```


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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R1 (master|MERGING)
$ git log --graph --all
* commit 4845a271388ce32039c213cf8e3d689b1db7a30e (test2)
  Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
  Date:   Wed Oct 18 21:19:25 2023 +0200

    otra actriz mas

* commit b67fad33c72f2df9ff5f24e667436963a1a4d30c (HEAD -> master)
  Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
  Date:   Wed Oct 18 21:16:31 2023 +0200

    Se añade una 4 actriz

* commit 39cb716935ebda6b75381220133b91209ac6dcb4 (test)
  Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
  Date:   Wed Oct 18 21:01:58 2023 +0200

    Tres actrices

* commit ac762d81edf01705dd1ad276ceac3cb84b2bdca2
  Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
  Date:   Wed Oct 18 20:55:35 2023 +0200

    Tres actores
```

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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2-remoto
$ git init --bare
Initialized empty Git repository in C:/Users/Jorge/Desktop/Curso-github/Practica1/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:

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1
$ git clone /S2R2-remoto S2R2
Cloning into 'S2R2'...
```

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

```
actores.txt  actrices.txt  directores.txt  salida.txt
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S2R2 > directores.txt
1 Tarantino
2 Martin Scorsese
3 Alejandro Gomez Monteverde|
```

b) Haz un commit.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (master)
$ git add directores.txt

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (master)
$ git commit -m "3 directores"
[master (root-commit) 480cb92] 3 directores
1 file changed, 3 insertions(+)
create mode 100644 directores.txt
```

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

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 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 C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2-remoto
 * [new branch]      master -> master

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (master)
$ git log --all
commit 480cb92dde4ffba22dc65252a9650102a6d9e063 (HEAD -> master, origin/master)
Author: Jorge Cabrera <jorcabort@alu.edu.gv.es>
Date:   Wed Oct 18 21:36:27 2023 +0200

    3 directores
```

d) Crea una rama denominada version1.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (master)
$ git branch version1

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (master)
$ git branch
* master
  version1
```

e) Cambia a la rama version1.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (master)
$ git checkout version1
Switched to branch 'version1'

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (version1)
$ git branch
  master
* 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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (version1)
$ git add directores.txt

Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (version1)
$ git commit -m "Añado dos directores mas"
[version1 799298e] Añado dos directores mas
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.

```
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date:   Wed Oct 18 21:41:15 2023 +0200

    Añado dos directores mas

commit 480cb92dde4ffba22dc65252a9650102a6d9e063 (origin/master, master)
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date:   Wed Oct 18 21:36:27 2023 +0200

    3 directores

jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1/S2R2 (version1)
$ git push
fatal: The current branch version1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin version1

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

En este paso no me deja subirlo con push esta vez no entiendo muy bien el porque...

```
MINGW64:/C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2
Practica1/ repo1/

jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github
$ cd Practica1

jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1
$ cd S2R2

jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2
(version1)
$ git push -u origin version1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2-remoto
 * [new branch]      version1 -> version1
branch 'version1' set up to track 'origin/version1'.

jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2
(version1)
```

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

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 ~/Desktop/Curso-github/Practica1
$ 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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R3
(master)
$ git log
commit 480cb92dde4ffba22dc65252a9650102a6d9e063 (HEAD -> master, origin/master,
origin/HEAD)
author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
date: Wed Oct 18 21:36:27 2023 +0200

    3 directores
```

b) Cambia a la rama version1. Muestra el resultado del comando.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R3
(master)
$ git branch
* master
```

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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R3
(master)
$ git branch version1

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R3
(master)
$ git branch
* master
  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.

```
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S2R3 > directores.txt
1 Tarantino
2 Martin Scorsese
3 Alejandro Gomez Monteverde
4 Stanley Kubrick
5 Alfred Hitchcock
```

d) Cambia a la rama master.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R3
(master)
$ git merge version1
Updating 480cb92..93381b8
Fast-forward
 directores.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

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

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R3
(master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2-remoto/
 480cb92..93381b8 master -> master

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R3
(master)
$ git log
commit 93381b8bd93648a7b1298c5c52b1c1942293be07 (HEAD -> master, origin/master,
origin/HEAD, version1)
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Wed Oct 18 22:02:12 2023 +0200

    5 directores

commit 480cb92dde4ffba22dc65252a9650102a6d9e063
Author: Jorge Cabrera <jorcabort@alu.edu.gva.es>
Date: Wed Oct 18 21:36:27 2023 +0200

    3 directores
```

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

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1
$ cd S2R2

Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2 (version1)
$
```

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.

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2
$ git fetch origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 260 bytes | 13.00 KiB/s, done.
From C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2-remoto
* branch      master      -> FETCH_HEAD
   480cb92..93381b8 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.

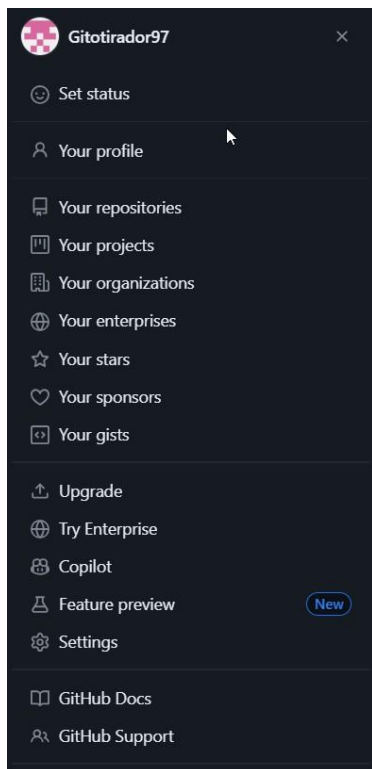
```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2
(version1)
$ git pull
Already up to date.
```

c) Comprueba que aparecen los 5 nombres de directores esperados

```
Jorge@DESKTOP-M8HVVJ6 MINGW64 /C:/Users/Jorge/Desktop/Curso-github/Practica1/S2R2
(version1)
$ cat directores.txt
Tarantino
Martin Scorsese
Alejandro Gomez Monteverde
Stanley Kubrick
Alfred Hitchcock
```

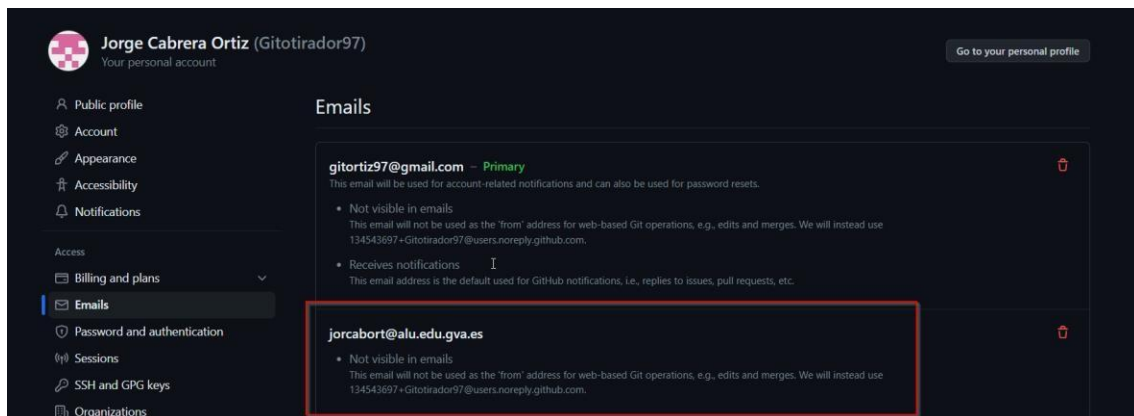
4. GitHub

1. Crea una cuenta en GitHub

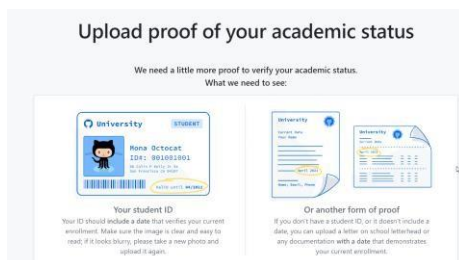


Ya tenia una cuenta de Git-hub

2. Añade tu dirección de correo de educación



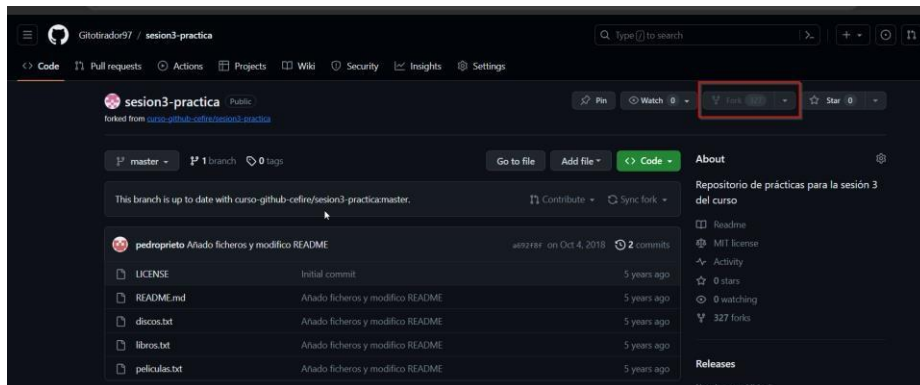
3. Solicita un descuento para uso educativo: <https://education.github.com/pack>



No tengo actualmente el carnet de estudiante del instituto.

4. Haz un fork del repositorio localizado en la siguiente url: <https://github.com/Gitotirador97/sesion3-practica>.

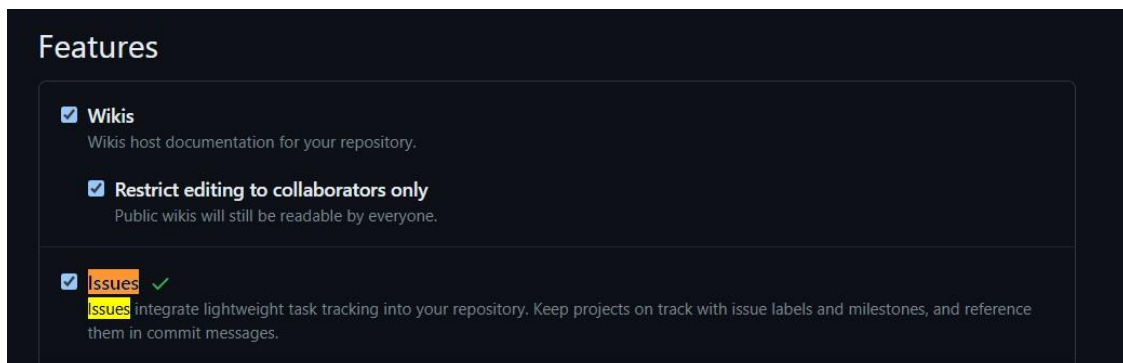
A partir de este momento todas las tareas que se indican se deben realizar en tu repositorio (el que has clonado mediante el fork).



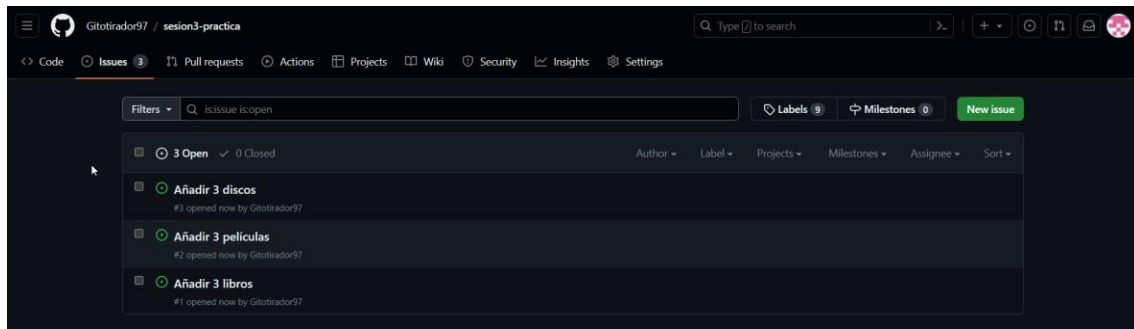
a) Realiza un primer commit para poner tu nombre y apellidos en el fichero README.md



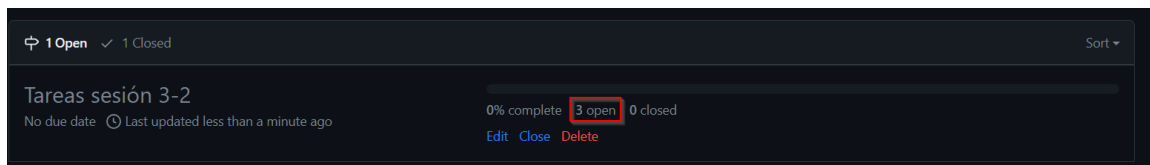
b) Crea 3 issues con los siguientes títulos. Si no ves la pestaña de issues, actívala desde los ajustes (settings) del repositorio.



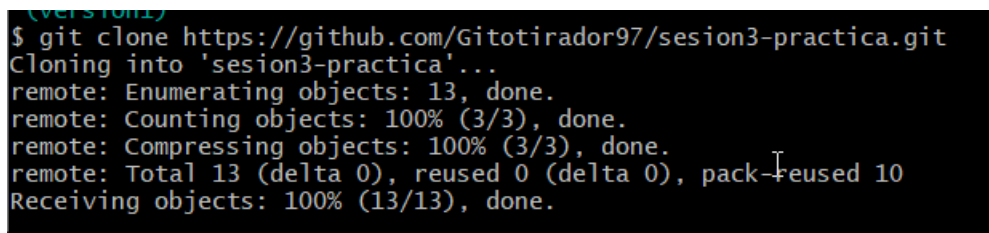
Añadir 3 libros, Añadir 3 películas, Añadir 3 discos



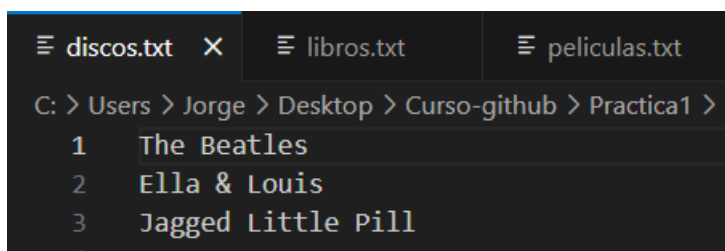
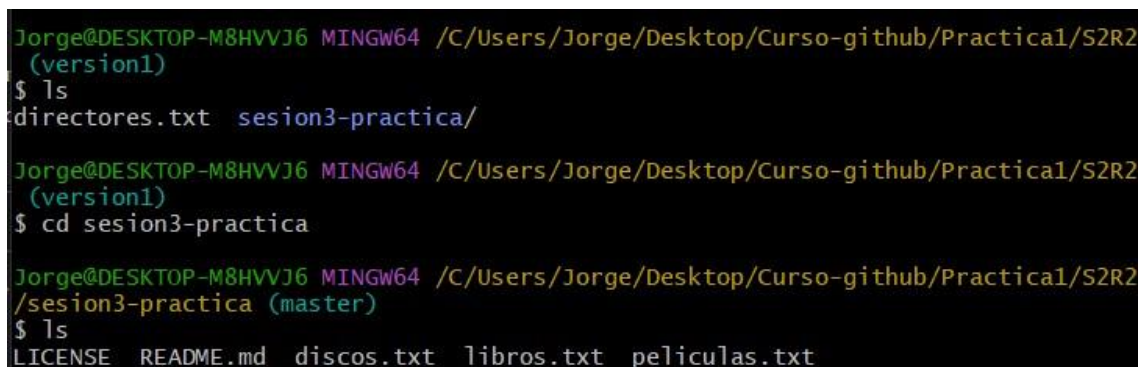
c) Crea una milestone denominada Tareas sesión 3-2 que contenga los 3 issues creados.



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.



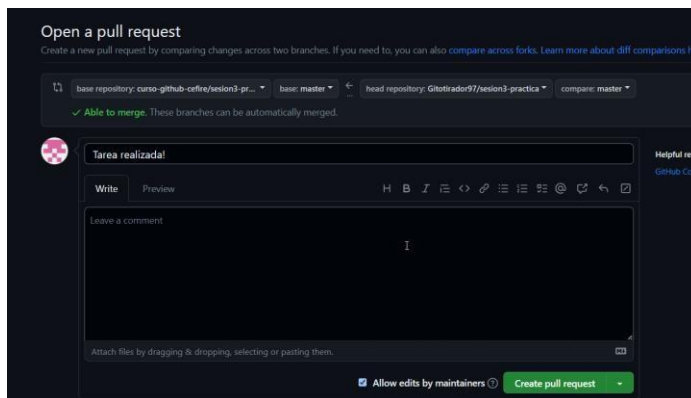
e) Haz una captura de pantalla de los comandos que has utilizado para hacer los commits y subir los cambios a GitHub.



```
discos.txt libros.txt peliculas.txt
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S
1 Juego de Tronos
2 El Juego de Ender
3 La Historia Interminable
4
```

```
discos.txt libros.txt peliculas.txt
C: > Users > Jorge > Desktop > Curso-github > Practica1 > S
1 Orgullo y Prejuicio
2 La boda de mi mejor amiga
3 Alta Fidelidad
4
```

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



5. Entrega de la tarea

La entrega de la tarea se realiza a través de la pull request en GitHub. No olvides incluir en la carpeta capturas todas las capturas de pantalla realizadas en la práctica (Fundamentos de Git I, Fundamentos de Git II y GitHub).

Todas las capturas realizadas estan en este PDF