

Fundamentos 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`.

```
rafa@DESKTOP-5PRARQG:~$ git --version
git version 2.30.2
rafa@DESKTOP-5PRARQG:~$
```

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

```
rafa@DESKTOP-5PRARQG:~$ git config --list
core.excludesfile=/home/rafa/.gitignore_global
user.name=Rafa Gandolfo Garcia
user.email=rafagandolfogarcia@gmail.com
rafa@DESKTOP-5PRARQG:~$
```

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

- Crea un repositorio Git.
- Crea un fichero denominado libros.txt. Añade tres títulos de libros cada uno en una línea distinta.
- Haz un primer commit.

```
rafa@DESKTOP-5PRARQG:~/S1R1$ touch libros.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ nano libros.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ car libros.txt
-bash: car: command not found
rafa@DESKTOP-5PRARQG:~/S1R1$ cat libros.txt
Historia de dos ciudades - Charles Dickens
El Conde de Montecristo - Alejandro Dumas
Harry Potter - J.K Rowling
rafa@DESKTOP-5PRARQG:~/S1R1$ git add .
rafa@DESKTOP-5PRARQG:~/S1R1$ git commit -m 'commit de libros'
[main (root-commit) 9fe0b3c] commit de libros
 1 file changed, 3 insertions(+)
 create mode 100644 libros.txt
rafa@DESKTOP-5PRARQG:~/S1R1$
```

d) Añade dos libros al archivo libros.txt.

```
rafa@DESKTOP-5PRARQG:~/S1R1$ cat libros.txt
Historia de dos ciudades - Charles Dickens
El Conde de Montecristo - Alejandro Dumas
Harry Potter - J.K Rowling
El Señor de los Anillos - Tolkien
Guerra y Paz - Tolstoi
```

e) Haz un segundo commit.

```
rafa@DESKTOP-5PRARQG:~/S1R1$ git add .
rafa@DESKTOP-5PRARQG:~/S1R1$ git commit -m 'segundo commit libros'
[main c013cd6] segundo commit libros
1 file changed, 2 insertions(+)
rafa@DESKTOP-5PRARQG:~/S1R1$
```

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

```
rafa@DESKTOP-5PRARQG:~/S1R1$ touch peliculas.txt && nano peliculas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ cat peliculas.txt
Hacia Rutas Salvajes
Terminator
Star Wars
```

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

```
rafa@DESKTOP-5PRARQG:~/S1R1$ git status
On branch main
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.

```
rafa@DESKTOP-5PRARQG:~/S1R1$ touch comidas.txt && nano comidas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ nano comidas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$
```

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

```

rafa@DESKTOP-5PRARQG:~/S1R1$ git add comidas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ git add peliculas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ git commit -m 'Comidas y peliculas'
[main 7f6c240] Comidas y peliculas
 2 files changed, 6 insertions(+)
 create mode 100644 comidas.txt
 create mode 100644 peliculas.txt

```

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

```

rafa@DESKTOP-5PRARQG:~/S1R1$ rm comidas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ ls
libros.txt  peliculas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$

```

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

```

rafa@DESKTOP-5PRARQG:~/S1R1$ nano peliculas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ cat peliculas.txt
Hacia Rutas Salvajes
Terminator
Star Wars
Regreso al futuro
Piratas del Caribe

```

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

```

rafa@DESKTOP-5PRARQG:~/S1R1$ git diff
diff --git a/comidas.txt b/comidas.txt
deleted file mode 100644
index 6a0e52a..0000000
--- a/comidas.txt
+++ /dev/null
@@ -1,3 +0,0 @@
-Pizza
-Espaguettis con pesto
-Kebab (falafel)
diff --git a/peliculas.txt b/peliculas.txt
index edbaa0e..ae00dbd 100644
--- a/peliculas.txt
+++ b/peliculas.txt
@@ -1,3 +1,5 @@
Hacia Rutas Salvajes
Terminator
Star Wars
+Regreso al futuro
+Piratas del Caribe

```

m) Añade los cambios al área de preparación.

```
rafa@DESKTOP-5PRARQG:~/S1R1$ git add .
rafa@DESKTOP-5PRARQG:~/S1R1$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    comidas.txt
        modified:   peliculas.txt
```

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.

```
rafa@DESKTOP-5PRARQG:~/S1R1$ git status
On branch main
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")
rafa@DESKTOP-5PRARQG:~/S1R1$
```

ñ) Haz un cuarto commit.

```
rafa@DESKTOP-5PRARQG:~/S1R1$ git commit -m 'cuarto commit'
[main 0af5bec] cuarto commit
2 files changed, 2 insertions(+), 3 deletions(-)
delete mode 100644 comidas.txt
```

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

```
delete mode 100644 comidas.txt
rafa@DESKTOP-5PRARQG:~/S1R1$ touch datos.bak && nano .
rafa@DESKTOP-5PRARQG:~/S1R1$ touch datos.bak && nano datos.bak
rafa@DESKTOP-5PRARQG:~/S1R1$ cat datos.bak
Diez negritos - Agatha Christie
El Cazador de Sueños - Stephen King
El Invierno del Mundo - Ken Follet
```

p) Crea una subcarpeta denominada output. Crea un archivo denominado salida.txt en su interior. Escribe tu nombre y apellidos en dicho archivo.

```
rafa@DESKTOP-5PRARQG:~/S1R1/output$ touch salida.txt && nano salida.txt
rafa@DESKTOP-5PRARQG:~/S1R1/output$ cat salida.txt
Rafa Gandolfo García
```

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

```
rafa@DESKTOP-5PRARQG:~/S1R1/output$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ../datos.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.

```
rafa@DESKTOP-5PRARQG:~/S1R1/output$ touch .gitignore && nano .gitignore
rafa@DESKTOP-5PRARQG:~/S1R1/output$ cat .gitignore
/output
*.bak
```

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.

```
rafa@DESKTOP-5PRARQG:~/S1R1$ git status
On branch main
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)
rafa@DESKTOP-5PRARQG:~/S1R1$
```

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

```
rafa@DESKTOP-5PRARQG:~/S1R1$ git add .gitignore
rafa@DESKTOP-5PRARQG:~/S1R1$ git commit -m 'commit del gitignore'
[main af3ea8d] commit del gitignore
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
```

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

```

rafa@DESKTOP-5PRARQG:~/S1R1$ git log
commit af3ea8d575d3369f00ccbdef9ac149a9fb70583b (HEAD -> main)
Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
Date:   Fri Sep 24 09:50:56 2021 +0200

    commit del gitignore

commit 0af5becc3c260cc69e800abd413c2bacc9fb82fa
Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
Date:   Fri Sep 24 09:28:47 2021 +0200

    cuarto commit

commit 7f6c240f5c2496858f56102e6c70fa76e942fb56
Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
Date:   Fri Sep 24 09:19:00 2021 +0200

    Comidas y peliculas

commit c013cd60a368f7f11512bbe6eb4d149be0768a84
Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
Date:   Fri Sep 24 09:08:07 2021 +0200

    segundo commit libros

commit 9fe0b3ceccf60f9afc3b86bc8b477848c171751a
Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
Date:   Fri Sep 24 09:03:51 2021 +0200

    commit de libros
:....skipping...
commit af3ea8d575d3369f00ccbdef9ac149a9fb70583b (HEAD -> main)
Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>

```

3. Fundamentos Git II

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

```

rafa@DESKTOP-5PRARQG:~$ cd S2RI
rafa@DESKTOP-5PRARQG:~/S2RI$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:     git config --global init.defaultBranch <name>
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:     git branch -m <name>
Initialized empty Git repository in /home/rafa/S2RI/.git/
rafa@DESKTOP-5PRARQG:~/S2RI$

```

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

```

rafa@DESKTOP-5PRARQG:~/S2RI$ touch actores.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ nano actores.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ cat actores.txt
cat: actores.txt: No such file or directory
rafa@DESKTOP-5PRARQG:~/S2RI$ cat actores.txt
Brad Pitt
Orlando Bloom
Meryl Streep

```

- c) Haz un primer commit.

```

rafa@DESKTOP-5PRARQG:~/S2RI$ git add actores.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ git commit -m 'commit de actores'
[master (root-commit) ac5bea6] commit de actores
1 file changed, 3 insertions(+)
create mode 100644 actores.txt
rafa@DESKTOP-5PRARQG:~/S2RI$

```

- d) Crea una rama denominada test.

```

rafa@DESKTOP-5PRARQG:~/S2RI$ git branch test

```

- e) Cambia a la rama test

```

rafa@DESKTOP-5PRARQG:~/S2RI$ 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.

```

rafa@DESKTOP-5PRARQG:~/S2RI$ touch actrices.txt && nano actrices.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ git add actrices.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ git commit -m 'commit de actrices'
[test c69eeeb] commit de actrices
1 file changed, 3 insertions(+)
create mode 100644 actrices.txt
rafa@DESKTOP-5PRARQG:~/S2RI$

```

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

```
rafa@DESKTOP-5PRARQG:~/S2RI$ git log --graph --all
* commit c69eeebbf6488400eb10acb9a8feb006900af3df (HEAD -> test)
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date:   Fri Sep 24 10:26:58 2021 +0200

    commit de actrices

* commit ac5bea6ac6d103000b65a3a57dec4339f99d16f1 (master)
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date:   Fri Sep 24 10:09:34 2021 +0200

    commit de actores
```

h) Cambia a la rama master.

```
rafa@DESKTOP-5PRARQG:~/S2RI$ git checkout master
Switched to branch 'master'
rafa@DESKTOP-5PRARQG:~/S2RI$
```

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.

```
rafa@DESKTOP-5PRARQG:~/S2RI$ git merge test
Updating ac5bea6..c69eeeb
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.

```
rafa@DESKTOP-5PRARQG:~/S2RI$ git branch test2
rafa@DESKTOP-5PRARQG:~/S2RI$
```

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


```

rafa@DESKTOP-5PRARQG:~/S2RI$ nano actrices.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ commit add actrices.txt
-bash: commit: command not found
rafa@DESKTOP-5PRARQG:~/S2RI$ git add actrices.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ git commit -m 'segundo commit actrices'
[master a486d6c] segundo commit actrices
1 file changed, 2 insertions(+)
rafa@DESKTOP-5PRARQG:~/S2RI$

```

l) Cambia a la rama test2

```

rafa@DESKTOP-5PRARQG:~/S2RI$ git checkout test2
Switched to branch 'test2'

```

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

```

rafa@DESKTOP-5PRARQG:~/S2RI$ nano actrices.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ cat actrices.txt
cat: actrices.txt: No such file or directory
rafa@DESKTOP-5PRARQG:~/S2RI$ cat actrices.txt
Jennifer Lawrence
Keira Knightley
Blanca Suarez
Anna Castillo

```

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

```

* commit a486d6ccd5f9425f7e9ac10fa5b1f0f88b8b9496 (master)
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date:   Fri Sep 24 10:36:03 2021 +0200

    segundo commit actrices

* commit c69eebbf6488400eb10acb9a8feb006900af3df (HEAD -> test2, test)
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date:   Fri Sep 24 10:26:58 2021 +0200

    commit de actrices

```

ñ) Cambia a la rama master

```

rafa@DESKTOP-5PRARQG:~/S2RI$ git commit -m 'tercer commit actrices rama test2'
[test2 794ace6] tercer commit actrices rama test2
1 file changed, 1 insertion(+)
rafa@DESKTOP-5PRARQG:~/S2RI$ git checkout master
Switched to branch 'master'
rafa@DESKTOP-5PRARQG:~/S2RI$

```

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.

```

rafa@DESKTOP-5PRARQG:~/S2RI$ git merge test2
Auto-merging actrices.txt
CONFLICT (content): Merge conflict in actrices.txt
Automatic merge failed; fix conflicts and then commit the result.
rafa@DESKTOP-5PRARQG:~/S2RI$ 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

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

no changes added to commit (use "git add" and/or "git commit -a")
rafa@DESKTOP-5PRARQG:~/S2RI$

```

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

```

rafa@DESKTOP-5PRARQG:~/S2RI$ git add actrices.txt
rafa@DESKTOP-5PRARQG:~/S2RI$ git commit -m 'commit conflicto'
[master e965033] commit conflicto
rafa@DESKTOP-5PRARQG:~/S2RI$ git merge test2
Already up to date.

```

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

```

* commit e9650335e6773fb4669408ffdeb96047981ae552 (HEAD -> master)
  Merge: a486d6c 794ace6
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date: Fri Sep 24 10:53:58 2021 +0200

    commit conflicto

* commit 794ace6a93290789aa3ebee6288be497459e26a (test2)
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date: Fri Sep 24 10:44:12 2021 +0200

    tercer commit actrices rama test2

* commit a486d6ccd5f9425f7e9ac10fa5b1f0f88b8b9496
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date: Fri Sep 24 10:36:03 2021 +0200

    segundo commit actrices

* commit c69eebbf6488400eb10acb9a8feb006900af3df (test)
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date: Fri Sep 24 10:26:58 2021 +0200

    commit de actrices

* commit ac5bea6ac6d103000b65a3a57dec4339f99d16f1
  Author: Rafa Gandolfo Garcia <rafagandolfogarcia@gmail.com>
  Date: Fri Sep 24 10:09:34 2021 +0200

    commit de 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.

```
rafa@DESKTOP-5PRARQG:~$ mkdir S2R2-remoto && cd S2R2-remoto
rafa@DESKTOP-5PRARQG:~/S2R2-remoto$ git init --bare
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/rafa/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:

```
rafa@DESKTOP-5PRARQG:~/S2R2$ git clone /home/rafa/S2R2-remoto
Cloning into 'S2R2-remoto'...
warning: You appear to have cloned an empty repository.
done.
rafa@DESKTOP-5PRARQG:~/S2R2$
```

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

```
rafa@DESKTOP-5PRARQG:~/S2R2$ touch directores.txt && nano directores.txt
rafa@DESKTOP-5PRARQG:~/S2R2$ cat directores.txt
Steven Spielberg
Robert Zemeckis
Alejandro Amenabar
```

b) Haz un commit.

```
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ git add directores.txt
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ git commit -m 'commit directores clonado'
[master (root-commit) 28ee99e] commit directores clonado
1 file changed, 3 insertions(+)
create mode 100644 directores.txt
```

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

```
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ git push origin mas
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 262 bytes | 262.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To /home/rafa/S2R2-remotoNuevo
* [new branch]      master -> master
```

d) Crea una rama denominada version1.

```
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ git branch version1
```

e) Cambia a la rama version1.

```
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ git checkout version1
Switched to branch 'version1'
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$
```

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.

```
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ nano directores.txt
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ cat directores.txt
Spielberg
Amenabar
Zemeckis

Almodovar
Columbus
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$
```

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.

```
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ git add directores.txt
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ git commit -m 'commit desde ramal directores'
[version1 cee673a] commit desde ramal directores
1 file changed, 3 insertions(+)
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$ git push origin master
Everything up-to-date
rafa@DESKTOP-5PRARQG:~/S2R2-nuevo/S2R2-remotoNuevo$
```

(este fue un push que hice erroneo, luego lo volvio a hacer pero poniendo la rama version1 en lugar de master, pero he borrado la terminal)

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

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

```
directores.txt
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ cat directores.txt
Spielberg
Amenabar
Zemeckis
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$
```

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.

```
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git checkout version1
Switched to branch 'version1'
Your branch is up to date with 'origin/version1'.
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$
```

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

```
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ cat directores.txt
Spielberg
Amenabar
Zemeckis

Almodovar
Columbus
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$
```

d) Cambia a la rama master.

```
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$
```

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

```
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git merge version1
Updating 28ee99e..cee673a
Fast-forward
 directores.txt | 3 +++
 1 file changed, 3 insertions(+)
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ cat directores.txt
Spielberg
Amenabar
Zemeckis

Almodovar
Columbus
```

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

```

rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To /home/rafa/S2R2-remotoNuevo
 28ee99e..cee673a master -> master
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ ls
directores.txt
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git add .
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git commit -m 'ultimo commit'
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git push origin master
Everything up-to-date
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$

```

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.

```

rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git remote add nombreRepo /home/rafa/S2R2-remotoNuevo
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git fetch nombreRepo
From /home/rafa/S2R2-remotoNuevo
 * [new branch]      master      -> nombreRepo/master
 * [new branch]      version1    -> nombreRepo/version1
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$

```

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.

```

rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git fetch nombreRepo
From /home/rafa/S2R2-remotoNuevo
 * [new branch]      master      -> nombreRepo/master
 * [new branch]      version1    -> nombreRepo/version1
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git pull nombreRepo master
int: Pulling without specifying how to reconcile divergent branches is
int: discouraged. You can squelch this message by running one of the following
int: commands sometime before your next pull:
int:
int:   git config pull.rebase false  # merge (the default strategy)
int:   git config pull.rebase true   # rebase
int:   git config pull.ff only       # fast-forward only
int:
int: You can replace "git config" with "git config --global" to set a default
int: preference for all repositories. You can also pass --rebase, --no-rebas
int: or --ff-only on the command line to override the configured default per
int: invocation.
From /home/rafa/S2R2-remotoNuevo
 * branch            master      -> FETCH_HEAD
Already up to date.

```

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

```
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ git fetch nombreRe  
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ ls  
directores.txt  
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$ cat directores.txt  
Spielberg  
Amenabar  
Zemeckis  
  
Almodovar  
Columbus  
rafa@DESKTOP-5PRARQG:~/S2R3-remoto/S2R2-remotoNuevo$
```