

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 --version

```
> git --version
git version 2.42.0
```

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.

```
[includeIf "gitdir:~/dev/work/"]
  path = ~/.config/git/config-work

[user]
  name = Sergio Lain
  email = sergio.lac@pm.me
  signingKey = 8429B2EE312F8150

[safe]
  directory = ~/dev

[credential "https://github.com"]
  username = Matt-FTW

[filter "lfs"]
  clean = git-lfs clean -- %f
  smudge = git-lfs smudge -- %f
  process = git-lfs filter-process
  required = true

[init]
  defaultBranch = main

[protocol "file"]
  allow = always

[credential]
  helper = libsecret

[core]
  editor = nvim
  pager = delta
  excludesfile = ~/.config/git/ignore
```

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

1. Crea un repositorio Git

```
~/Repos/Class
> mkdir S1R1

~/Repos/Class
> cd S1R1/

Repos/Class/S1R1
> git init
Inicializado repositorio Git vacío en /home/matt/Repos/Class/S1R1/.git/

S1R1 on ✘ main
> l
Permissions Size User Date Modified Git Name
drwxr-xr-x - matt 13 sep 16:35 -I .git
```

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

```
S1R1 on ✘ main
> vim libros.txt

S1R1 on ✘ main [?] took 56s
> cat libros.txt
-----
File: libros.txt
-----
1 La Bilia
2 El Señor de los Anillos
3 El Código Da Vinci
```

3. Haz un primer *commit*.

```
S1R1 on ✘ main [?]
> git add libros.txt

S1R1 on ✘ main [*]
> git status
En la rama main

No hay commits todavía

Cambios a ser confirmados:
(usa "git rm --cached <archivo>..." para sacar del área de stage)
    nuevos archivos: libros.txt

S1R1 on ✘ main [*]
> git commit -m "Initial Commit"
[main (commit-raíz) 9eaaea8] Initial Commit
  1 file changed, 3 insertions(+)
  create mode 100644 libros.txt
```

4. Añade dos libros al archivo libros.txt.

```
S1R1 on ♫ main took 27s
> vim libros.txt

S1R1 on ♫ main [!] took 21s
> cat libros.txt

+-----+
| File: libros.txt |
+-----+
| 1 | La Bilia
| 2 | El Señor de los Anillos
| 3 | El Código Da Vinci
| 4 | + El Diario de Ana Frank
| 5 | + El Alquimista
+-----+
```

5. Haz un segundo *commit*.

```
S1R1 on ♫ main [!]
> git add libros.txt

S1R1 on ♫ main [+]
> git commit -m "2 New Books Added"
[main 66d5054] 2 New Books Added
  1 file changed, 2 insertions(+)
```

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

```
S1R1 on ♫ main
> vim peliculas.txt

S1R1 on ♫ main [?] took 5m47s
> cat peliculas.txt

+-----+
| File: peliculas.txt |
+-----+
| 1 | Oppenheimer
| 2 | Super Mario: The Movie
| 3 | Avengers: Infinity War
+-----+
```

7. Haz una captura de pantalla del comando `git status`.

```
SIR1 on ♫ main [?]
> git status
En la rama main
Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que será confirmado)
    peliculas.txt

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
```

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

```
SIR1 on ♫ main [?]
> vim comidas.txt

SIR1 on ♫ main [?] took 2m17s
> cat comidas.txt

+-----+
|          File: comidas.txt
+-----+
| 1 | Patatas
| 2 | Patatas de Chocolate
| 3 | Patatas Fritas
+-----+
```

9. Haz un tercer *commit* que incluya los archivos peliculas.txt y comidas.txt.

```
SIR1 on ♫ main [?]
> git add .

SIR1 on ♫ main [+]
> git status
En la rama main
Cambios a ser confirmados:
  (usa "git restore --staged <archivo>..." para sacar del área de stage)
    nuevos archivos: comidas.txt
    nuevos archivos: peliculas.txt

SIR1 on ♫ main [+]
> git commit -m "New files: Peliculas.txt and Comidas.txt"
[main 56a2f21] New files: Peliculas.txt and Comidas.txt
  2 files changed, 6 insertions(+)
  create mode 100644 comidas.txt
  create mode 100644 peliculas.txt
```

10. Elimina el archivo comidas.txt desde el navegador de archivos.

```
SIR1 on ⚡ main
> rm comidas.txt

SIR1 on ⚡ main [x]
> l
Permissions Size User Date Modified Git Name
drwxr-xr-x    - matt 14 sep 08:42   -I .git
.rw-r--r--    91 matt 14 sep 08:40   -- libro.txt
.rw-r--r--    58 matt 14 sep 08:41   -- peliculas.txt
```

11. Añade dos películas más al archivo peliculas.txt.

```
SIR1 on ⚡ main [x]
> vim peliculas.txt

SIR1 on ⚡ main [x!] took 26s
> cat peliculas.txt
-----| File: peliculas.txt
-----|-----|-----|
1      | Oppenheimer
2      | Super Mario: The Movie
3      | Avengers: Infinity War
4 +    | El Padrino
5 +    | Holocausto Canibal
```

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

```
SIR1 on ⚡ main [x!]
> git diff
-----|-----|
deleted: comidas.txt
-----|-----|
@ comidas.txt:1 @
Patatas
Patatas de Chocolate
Patatas Fritas
-----|-----|
modified: peliculas.txt
-----|-----|
@ peliculas.txt:1 @
Oppenheimer
Super Mario: The Movie
Avengers: Infinity War
El Padrino
Holocausto Canibal
```

13. Añade los cambios al área de preparación.

```
SIR1 on ⚡ main [x!]
> git add _
```

14. 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.

```
SIR1 on ⚡ main [x+]
> git status
En la rama main
Cambios a ser confirmados:
  (usa "git restore --staged <archivo>..." para sacar del área de stage)
    borrados:      comidas.txt
    modificados:   peliculas.txt
```

15. Haz un cuarto commit.

```
SIR1 on ⚡ main [x+]
> git commit -m "Comidas.txt was deleted; added 2 more movies to Peliculas.txt"
[main 3b6a07c] Comidas.txt was deleted; added 2 more movies to Peliculas.txt
  2 files changed, 2 insertions(+), 3 deletions(-)
  delete mode 100644 comidas.txt
```

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

```
SIR1 on ⚡ main [?]
> vim datos.bak

SIR1 on ⚡ main [?]
> cat datos.bak
+-----+
|                               File: datos.bak |
+-----+
| 1 | El Principito
| 2 | Carrie
| 3 | La Clave esta en Rebeca
+-----+
```

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

```
SIR1 on ♫ main [?]
> mkdir output

SIR1 on ♫ main [?]
> cd output/

SIR1/output on ♫ main [?] took 8s
> vim salida.txt

SIR1/output on ♫ main [?] took 8s
> cat salida.txt

+-----+
|           File: salida.txt |
+-----+
| 1       Sergio Laín Camús |
+-----+
```

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

```
SIR1 on ♫ main [?]
> git status
En la rama main
Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que será confirmado)
    datos.bak
    output/
```

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

```
SIR1 on ♫ main [?]
> vim .gitignore

SIR1 on ♫ main [?] took 9s
> cat .gitignore

+-----+
|           File: .gitignore |
+-----+
| 1       *.bak |
| 2       output/ |
+-----+
```

20. 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.

```
SIR1 on ♫ main [?]
› git status
En la rama main
Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que será confirmado)
    .gitignore
```

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

```
SIR1 on ♫ main [?]
› git add .gitignore

SIR1 on ♫ main [+]
› git commit -m ".gitignore added to the repo"
[main 2a76e48] .gitignore added to the repo
  1 file changed, 2 insertions(+)
  create mode 100644 .gitignore
```

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

```
> | < 5/5 -s
> * 2a76e48 (HEAD -> main) .gitignore added to the repo hace 76 segundos
* 3b6a07c Comidas.txt was deleted; added 2 more movies to Peliculas.txt hace 8 minutos
* 56a2f21 New files: Peliculas.txt and Comidas.txt hace 14 minutos
* 74274e9 2 New Books Added hace 16 minutos
* 9b89edc Initial Commit hace 17 minutos
```

```
SIR1 on ♫ main
› git log
commit 2a76e48026bcfc0cead647f59c10d3cf4f7d3d0e (HEAD -> main)
Author: Sergio Laín <sergiolain11@gmail.com>
Date:   Thu Sep 14 08:55:27 2023 +0200

  .gitignore added to the repo

commit 3b6a07c76c55cee84188a8ed790434833a3e5047
Author: Sergio Laín <sergiolain11@gmail.com>
Date:   Thu Sep 14 08:48:23 2023 +0200

  Comidas.txt was deleted; added 2 more movies to Peliculas.txt

commit 56a2f21ae8fca122df165e6b7ce55d5fc85546eb
Author: Sergio Laín <sergiolain11@gmail.com>
Date:   Thu Sep 14 08:42:55 2023 +0200

  New files: Peliculas.txt and Comidas.txt

commit 74274e92a4d4a6552774faa701b2432450a4d3ae
Author: Sergio Laín <sergiolain11@gmail.com>
Date:   Thu Sep 14 08:41:03 2023 +0200

  2 New Books Added

commit 9b89edc85580a986575225f3360df1bdab33e69f
Author: Sergio Laín <sergiolain11@gmail.com>
Date:   Thu Sep 14 08:39:27 2023 +0200

  Initial Commit
```

Fundamentos de Git II

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

```
Repos/Class/DAW
> mkdir S2R1

Repos/Class/DAW
> cd S2R1/

Class/DAW/S2R1
> git init
Inicializado repositorio Git vacío en /home/matt/Repos/Class/DAW/S2R1/.git/
```

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

```
S2R1 on ♫ main [?]
> vim actores.txt

S2R1 on ♫ main [?]
> cat actores.txt
```

File: actores.txt	
1	Brad Pitt
2	Meryl Streep
3	Cary Grant

- Haz un primer commit.

```
S2R1 on ♫ main [?]
> git add actores.txt

S2R1 on ♫ main [+]
> git commit -m "Initial Commit"
[main (commit-raíz) d3e704a] Initial Commit
 1 file changed, 3 insertions(+)
  create mode 100644 actores.txt
```

- Crea una rama denominada test.

```
S2R1 on ♫ main took 24s
> git branch test

S2R1 on ♫ main
> git branch
* main
  test
```

5. Cambia a la rama test

```
S2R1 on ⚡ main
❯ git switch test
Cambiado a rama 'test'

S2R1 on ⚡ test
❯ git branch
  main
* test
```

```
S2R1 on ⚡ main
❯ git checkout test
Cambiado a rama 'test'

S2R1 on ⚡ test
❯ git branch
  main
* test
```

O

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

```
S2R1 on ⚡ test
❯ vim actrices.txt

S2R1 on ⚡ test [?] took 55s
❯ cat actrices.txt

|   | File: <b>actrices.txt</b> |
|---|---------------------------|
| 1 | Scarlett Johansson        |
| 2 | Nicole Kidman             |
| 3 | Natalie Portman           |



S2R1 on ⚡ test [?]
❯ git add actrices.txt

S2R1 on ⚡ test [+]
❯ git commit -m "New file added: Actrices.txt"
[test 87edc0d] New file added: Actrices.txt
  1 file changed, 3 insertions(+)
   create mode 100644 actrices.txt
```

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

```
S2R1 on ⚡ test
❯ git log --graph --all
* commit 87edc0d09f62f23bb0334362d7c174b07e6d9dcc (HEAD -> test)
| Author: Sergio Lain <sergiolain11@gmail.com>
| Date:   Thu Sep 14 09:19:12 2023 +0200
|
|   New file added: Actrices.txt
|
* commit d3e704acdd92703c91416b1b3a8c0a76dcc4fd89 (main)
| Author: Sergio Lain <sergiolain11@gmail.com>
| Date:   Thu Sep 14 09:12:38 2023 +0200
|
|   Initial Commit
|
> | < 2/2 -S —
> * 87edc0d (HEAD -> test) New file added: Actrices.txt hace 83 segundos
* d3e704a (main) Initial Commit hace 8 minutos
```

8. Cambia a la rama master.

```
S2R1 on ♫ test
> git checkout main
Cambiado a rama 'main'
```

9. 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.

```
S2R1 on ♫ main
> git merge test
Actualizando d3e704a..87edc0d
Fast-forward
  actrices.txt | 3 +++
  1 file changed, 3 insertions(+)
  create mode 100644 actrices.txt

S2R1 on ♫ main
> l
Permissions Size User Date Modified Git Name
drwxr-xr-x   - matt 14 sep 09:22  -I .git
.rw-r--r--  34 matt 14 sep 09:11  -- actores.txt
.rw-r--r--  49 matt 14 sep 09:22  -- actrices.txt

S2R1 on ♫ main
> git status
En la rama main
nada para hacer commit, el árbol de trabajo está limpio
```

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

```
S2R1 on ♫ main
> git branch test2
```

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

```
S2R1 on ♫ main
> vim actrices.txt

S2R1 on ♫ main [!]
> cat actrices.txt
+-----+
| File: actrices.txt
+-----+
| 1 | Scarlett Johansson
| 2 | Nicole Kidman
| 3 | Natalie Portman
| 4 +| Jennifer Lawrence
+-----+

S2R1 on ♫ main [!]
> git add actrices.txt

S2R1 on ♫ main [*]
> git commit -m "New actress added"
[main f8eab7] New actress added
  1 file changed, 1 insertion(+)
```

12. Cambia a la rama test2

```
S2R1 on ♫ main
› git checkout test2
Cambiado a rama 'test2'
```

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

```
S2R1 on ♫ test2
› vim actrices.txt

S2R1 on ♫ test2 [!] took 25s
› cat actrices.txt
+-----+
|          File: actrices.txt
+-----+
| 1 | Scarlett Johansson
| 2 | Nicole Kidman
| 3 | Natalie Portman
| 4 +| Emma Stone
+-----+


S2R1 on ♫ test2 [!]
› git add actrices.txt

S2R1 on ♫ test2 [+]
›

S2R1 on ♫ test2 [+]
› git commit -m "New actress added"
[test2 6179bf0] New actress added
 1 file changed, 1 insertion(+)
```

14. 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.

```
S2R1 on ⚡ test2
› git log --graph --all
* commit 6179bf0a29404c24d6b2488a7206c5cc74a2b105 (HEAD -> test2)
| Author: Sergio Laín <sergiolain11@gmail.com>
| Date:   Thu Sep 14 09:27:54 2023 +0200
|
|       New actress added
|
| * commit f8aeab7f2660f8c58c2705899d4310ea9f91c9bf (main)
| / Author: Sergio Laín <sergiolain11@gmail.com>
| Date:   Thu Sep 14 09:25:18 2023 +0200
|
|       New actress added
|
* commit 87edc0d09f62f23bb0334362d7c174b07e6d9dcc (test)
| Author: Sergio Laín <sergiolain11@gmail.com>
| Date:   Thu Sep 14 09:19:12 2023 +0200
|
|       New file added: Actrices.txt
|
* commit d3e704acdd92703c91416b1b3a8c0a76dcc4fd89
Author: Sergio Laín <sergiolain11@gmail.com>
Date:   Thu Sep 14 09:12:38 2023 +0200
|
Initial Commit
```

15. Cambia a la rama master

```
S2R1 on ⚡ test2
› git checkout main
Cambiado a rama 'main'
```

16. 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.

```
S2R1 on ⚡ main
› git merge test2
Auto-fusionando actrices.txt
CONFLICTO (contenido): Conflicto de fusión en actrices.txt
Fusión automática falló; arregle los conflictos y luego realice un commit con el resultado.

S2R1 on ⚡ main (MERGING) [=]
› git status
En la rama main
Tienes rutas no fusionadas.
  (arregla los conflictos y ejecuta "git commit"
  (usa "git merge --abort" para abortar la fusión)

Rutas no fusionadas:
  (usa "git add <archivo>..." para marcar una resolución)
    modificados por ambos:  actrices.txt

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
```

17. Resuelve el conflicto incorporando los dos nombres de actrices.

~/Repos/Class/DAW/S2R1	OURS (Current changes) f8aeab7f26 (HEAD -> main)	THEIRS (Incoming changes) 6179bf0a29												
Help: g?	1 Scarlett Johansson 2 Nicole Kidman 3 Natalie Portman 4 Jennifer Lawrence	1 Scarlett Johansson 2 Nicole Kidman 3 Natalie Portman 4 Emma Stone 5 Jennifer Lawrence												
Conflicts (1) U actrices.txt ✓	-----	-----												
Changes (4) ? actrices_BACKUP_55532.txt ? actrices_BASE_55532.txt ? actrices_LOCAL_55532.txt ? actrices_REMOTE_55532.txt														
<pre>› cat actrices.txt</pre> <table border="1"> <thead> <tr> <th colspan="2">File: actrices.txt</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Scarlett Johansson</td> </tr> <tr> <td>2</td> <td>Nicole Kidman</td> </tr> <tr> <td>3</td> <td>Natalie Portman</td> </tr> <tr> <td>4</td> <td>Emma Stone</td> </tr> <tr> <td>5</td> <td>Jennifer Lawrence</td> </tr> </tbody> </table>			File: actrices.txt		1	Scarlett Johansson	2	Nicole Kidman	3	Natalie Portman	4	Emma Stone	5	Jennifer Lawrence
File: actrices.txt														
1	Scarlett Johansson													
2	Nicole Kidman													
3	Natalie Portman													
4	Emma Stone													
5	Jennifer Lawrence													

18. 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.

```
S2R1 on ⚡ main [?] took 51s
❯ git log --graph --all
* commit 498ab88cff8d8b609b8d61f0b43a8587dc6349b2 (HEAD -> main)
| \
| Merge: f8aeab7 6179bf0
| | Author: Sergio Laín <sergiolain11@gmail.com>
| | Date:   Thu Sep 14 09:41:27 2023 +0200
|
|     Merge branch 'test2'
|
* commit 6179bf0a29404c24d6b2488a7206c5cc74a2b105 (test2)
| | Author: Sergio Laín <sergiolain11@gmail.com>
| | Date:   Thu Sep 14 09:27:54 2023 +0200
|
|     New actress added
|
* commit f8aeab7f2660f8c58c2705899d4310ea9f91c9bf
| / Author: Sergio Laín <sergiolain11@gmail.com>
| | Date:   Thu Sep 14 09:25:18 2023 +0200
|
|     New actress added
|
* commit 87edc0d09f62f23bb0334362d7c174b07e6d9dcc (test)
| | Author: Sergio Laín <sergiolain11@gmail.com>
| | Date:   Thu Sep 14 09:19:12 2023 +0200
|
|     New file added: Actrices.txt
|
* commit d3e704acdd92703c91416b1b3a8c0a76dcc4fd89
| | Author: Sergio Laín <sergiolain11@gmail.com>
| | Date:   Thu Sep 14 09:12:38 2023 +0200
|
|     Initial Commit
```

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

```
Repos/Class/DAW
> mkdir S2R2-remoto

Repos/Class/DAW
> cd S2R2-remoto/

Class/DAW/S2R2-remoto
> git init --bare
Inicializado repositorio Git vacío en /home/matt/Repos/Class/DAW/S2R2-remoto/
```

3. Clona el repositorio S2R2-remoto en una carpeta denominada S2R2. Adjunta captura de pantalla del resultado del comando de clonado.

```
Repos/Class/DAW
> git clone S2R2-remoto S2R2
Clonando en 'S2R2'...
advertencia: Pareces haber clonado un repositorio sin contenido.
hecho.

Repos/Class/DAW
> l
Permissions Size User Date Modified Name
drwxr-xr-x   - matt 14 sep 09:43  S2R1
drwxr-xr-x   - matt 14 sep 09:57  S2R2
drwxr-xr-x   - matt 14 sep 09:49  S2R2-remoto
drwxr-xr-x   - matt 14 sep 08:54  SIR1
```

A continuación realiza las siguientes acciones en el repositorio S2R2:

1. Crea un archivo denominado directores.txt. Añade el nombre de tres directores de cine.

```
S2R2 on ɻ main
> vim directores.txt

S2R2 on ɻ main [?] took 33s
> cat directores.txt
```

	File: directores.txt
1	Martin Scorsese
2	Steven Spielberg
3	Quentin Tarantino

2. Haz un *commit*.

```
S2R2 on ⚡ main [?]
› git add directores.txt

S2R2 on ⚡ main [+]
› git commit -m "Initial Commit"
[main (commit-raíz) 038924f] Initial Commit
  1 file changed, 3 insertions(+)
   create mode 100644 directores.txt
```

3. Realiza un *push* al repositorio remoto. Adjunta captura de pantalla del resultado.

```
S2R2 on ⚡ main
› git push
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Compresión delta usando hasta 24 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 927 bytes | 927.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To /home/matt/Repos/Class/DAW/S2R2-remoto
 * [new branch]      main -> main
```

4. Crea una rama denominada *version1*.

```
S2R2 on ⚡ main
› git branch version1

S2R2 on ⚡ main
› git branch
* main
  version1
```

5. Cambia a la rama *version1*.

```
S2R2 on ⚡ main
› git checkout version1
Cambiado a rama 'version1'
```

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

```
S2R2 on ♫ version1
> vim directores.txt

S2R2 on ♫ version1 [!] took 19s
> cat directores.txt

```

	File: directores.txt
1	Martin Scorsese
2	Steven Spielberg
3	Quentin Tarantino
4 +	Christopher Nolan
5 +	David Fincher


```
S2R2 on ♫ version1 [!]
> git add directores.txt

S2R2 on ♫ version1 [+]
> git commit -m "2 new directors"
[version1 5f25874] 2 new directors
  1 file changed, 2 insertions(+)
```

7. 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.

```
S2R2 on ♫ version1
> git push -u origin version1
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 24 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 996 bytes | 996.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To /home/matt/Repos/Class/DAW/S2R2-remoto
 * [new branch]      version1 -> version1
rama 'version1' configurada para rastrear 'origin/version1'.
```

4. Clona el repositorio S2R2-remoto en una segunda carpeta denominada S2R3.

```
Repos/Class/DAW
> git clone S2R2-remoto S2R3
Clonando en 'S2R3'...
hecho.
```

Realiza las siguientes acciones sobre ella:

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

```
S2R3 on 🌐 main
> ls
Permissions Size User Date Modified Git Name
drwxr-xr-x    - matt 14 sep 10:08  -I .git
.rw-r--r--    51 matt 14 sep 10:08  -- directores.txt

S2R3 on 🌐 main
> cat directores.txt
File: directores.txt
1 Martin Scorsese
2 Steven Spielberg
3 Quentin Tarantino
```

- 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.

```
S2R3 on 🌐 main
> git checkout version1
rama 'version1' configurada para rastrear 'origin/version1'.
Cambiado a nueva rama 'version1'

S2R3 on 🌐 version1
> git branch
  main
* version1
```

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

```
S2R3 on 🌐 version1
> cat directores.txt
File: directores.txt
1 Martin Scorsese
2 Steven Spielberg
3 Quentin Tarantino
4 Christopher Nolan
5 David Fincher
```

4. Cambia a la rama master.

```
S2R3 on ↵ version1
› git checkout main
Cambiado a rama 'main'
Tu rama está actualizada con 'origin/main'.
```

5. Incorpora los cambios de la rama version1 a la rama master.

```
S2R3 on ↵ main
› git merge version1
Actualizando 038924f..5f25874
Fast-forward
directores.txt | 2 ++
 1 file changed, 2 insertions(+)

S2R3 on ↵ main [↑]
› git status
En la rama main
Tu rama está adelantada a 'origin/main' por 1 commit.
  (usa "git push" para publicar tus commits locales)

nada para hacer commit, el árbol de trabajo está limpio
```

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

```
S2R3 on ↵ main [↑]
› git push
Total 0 (delta 0), reusados 0 (delta 0), pack-reusados 0
To /home/matt/Repos/Class/DAW/S2R2-remoto
  038924f..5f25874  main -> main

S2R3 on ↵ main
› git log --graph --all
* commit 5f25874e5effea01f1e9f7ec2a75c5358e49a9a7 (HEAD -> main, origin/version1, origin/main, origin/HEAD, version1)
| Author: Sergio Laín <sergiolain11@gmail.com>
| Date:   Thu Sep 14 10:05:25 2023 +0200
|
|     2 new directors
|
* commit 038924f8dd1a6e718f56ef002c8969fd4a41320e
| Author: Sergio Laín <sergiolain11@gmail.com>
| Date:   Thu Sep 14 09:59:36 2023 +0200
|
|     Initial Commit
```

5. Vuelve de nuevo a la carpeta S2R2 y realiza las siguientes acciones:
 1. Obtén los cambios que hay en el repositorio remoto **sin fusionarlos en la rama local**. Adjunta captura de pantalla del resultado del comando utilizado.

```
S2R2 on ⬆ main [↓]
> git fetch

S2R2 on ⬆ main [↓]
> git status
En la rama main
Tu rama está detrás de 'origin/main' por 1 commit, y puede ser avanzada rápido.
  (usa "git pull" para actualizar tu rama local)

nada para hacer commit, el árbol de trabajo está limpio
```

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

```
S2R2 on ⬆ main [↓]
> git pull
Actualizando 038924f..5f25874
Fast-forward
  directores.txt | 2 ++
    1 file changed, 2 insertions(+)

S2R2 on ⬆ main
> git status
En la rama main
Tu rama está actualizada con 'origin/main'.

nada para hacer commit, el árbol de trabajo está limpio
```

3. Comprueba que aparecen los 5 nombres de directores esperados.

File: directores.txt	
1	Martin Scorsese
2	Steven Spielberg
3	Quentin Tarantino
4	Christopher Nolan
5	David Fincher

Github

1. Crea una cuenta en GitHub

Te pongo la cuenta que ya tenia creada donde le asigno luego el email de clase.

The screenshot shows a GitHub profile for a user named 'Matt'. The profile picture is a portrait of a person with long brown hair. The bio reads: 'Music, Art, Videogames and Cookies lover. Drummer, pianist and music composer. Arch and Catpuccin enthusiast.' Below the bio, there's an 'Edit profile' button. The stats section shows 22 followers and 22 following. Under 'Achievements', there are several small circular icons representing different milestones. The 'Emails' section lists three email addresses:

- sergio.lac@pm.me** - Primary: Described as the default used for GitHub notifications, i.e., replies to issues, pull requests, etc.
- sergio.lac-misc@proton.me**: Described as visible in emails and used for web-based Git operations.
- sergiolain11@gmail.com**: Described as visible in emails and used for account-related notifications and password resets.

2. Añade tu dirección de correo de educación

The screenshot shows the 'Public email' settings in GitHub. It displays the email address **serlaicam@alu.edu.gva.es** and includes a 'Remove' button. A note at the bottom states: 'You can manage verified email addresses in your email settings'.

The screenshot shows the 'Emails' settings in GitHub. It lists the email address **sergiolain11@gmail.com** as Primary, described as visible in emails and used for account-related notifications and password resets. It also lists the email address **serlaicam@alu.edu.gva.es**, described as visible in emails and used for web-based Git operations.

3. Solicita un descuento para uso educativo (ver enlace en apartado de Referencias)

Ya lo tengo

Your recurring charges & discounts

GitHub Pro	\$4.00/month	On Sep 30, 2023
GitHub Coupon	\$4.00 off / month	Expires Sep 30, 2023

* Discount applies for GitHub Pro

Current plan

GitHub Pro Edit ▾

- ∞ Unlimited public/private repos
- ∞ Unlimited collaborators
- ✓ 3,000 Actions minutes/month
- ✓ 2GB of Packages storage
- ✓ 180 core-hours of Codespaces compute
- ✓ 20GB of Codespaces storage
- ✓ Code owners, required reviewers, GitHub Pages, and more advanced tools.

\$4 per month **\$0.00**

Trabajo con repositorios, issues, forks y pull requests

1. Haz un *fork* del repositorio localizado en la siguiente url: <https://github.com/curso-github-cefie/sesion3-practica>.

sesion3-practica Public
forked from curso-github-cefie/sesion3-practica

This branch is up to date with [curso-github-cefie/sesion3-practica@master](#).

Commit	Author	Message	Date	Commits
pedroprieto	Añado ficheros y modiflico README	a692fbf · 6 years ago	2 Commits	
LICENSE	Initial commit	6 years ago		
README.md	Añado ficheros y modiflico README	6 years ago		
discos.txt	Añado ficheros y modiflico README	6 years ago		
libros.txt	Añado ficheros y modiflico README	6 years ago		
peliculas.txt	Añado ficheros y modiflico README	6 years ago		

About
Repository de prácticas para la sesión 3 del curso

Readme
MIT license
Activity
0 stars
0 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Uso de GitHub en la práctica docente

Práctica 3.2

Repository de prácticas para la sesión 3 del curso

Incluye a continuación tu nombre y apellidos:

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

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

```
sesion3-practica on ♫ master [!]
> vim README.md

sesion3-practica on ♫ master [!] took 14s
> cat README.md
+-----+
| File: README.md
+-----+
| 1   # Uso de GitHub en la práctica docente
| 2 +
| 3   ## Práctica 3.2
| 4 +
| 5   Repositorio de prácticas para la sesión 3 del curso
| 6
| 7 ~ Incluye a continuación tu nombre y apellidos: Sergio Laín Camús
+-----+


sesion3-practica on ♫ master [!]
> git add README.md

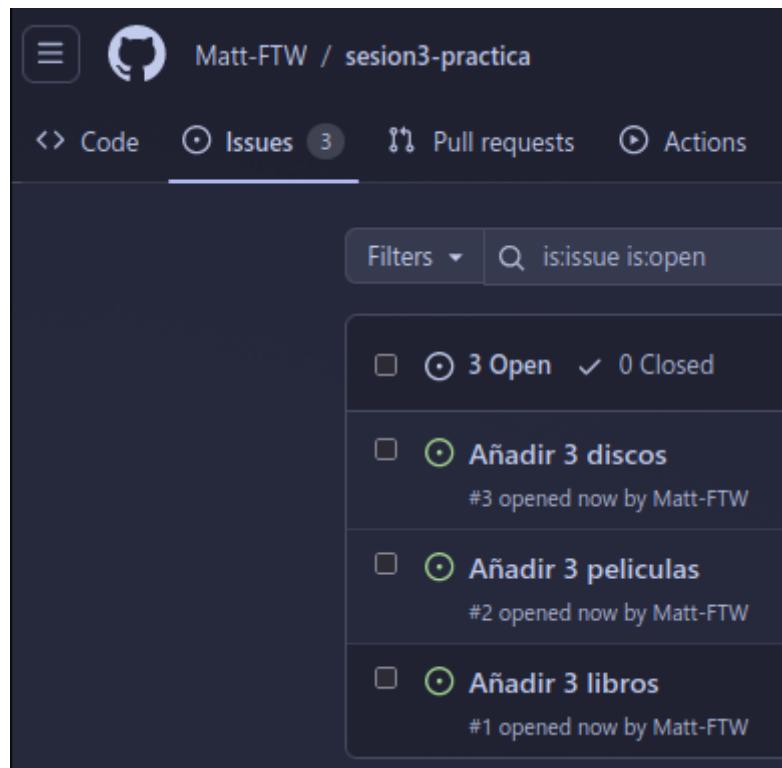
sesion3-practica on ♫ master [+]
> git commit -m "Name added to the README"
[master 587b243] Name added to the README
 1 file changed, 3 insertions(+), 1 deletion(-)

sesion3-practica on ♫ master [↑]
> git status
En la rama master
Tu rama está adelantada a 'origin/master' por 1 commit.
  (usa "git push" para publicar tus commits locales)

nada para hacer commit, el árbol de trabajo está limpio

sesion3-practica on ♫ master [↑]
> git push
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 24 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 1005 bytes | 1005.00 KiB/s, listo.
Total 3 (delta 2), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:Matt-FTW/sesion3-practica.git
  a692f8f..587b243  master -> master
```

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

The screenshot shows a GitHub repository named 'Matt-FTW / sesion3-practica'. The 'Issues' tab is selected, displaying three open issues:

- #3 Añadir 3 discos (opened 2 minutes ago by Matt-FTW)
- #2 Añadir 3 películas (opened 3 minutes ago by Matt-FTW)
- #1 Añadir 3 libros (opened 3 minutes ago by Matt-FTW)

The repository has 3 open issues and 0 closed issues.

4. 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.

```
sesion3-practica on ✘ master
> vim

sesion3-practica on ✘ master [!] took 50s
> cat peliculas.txt
+-----+
| File: peliculas.txt |
+-----+
| 1 | Orgullo y Prejuicio
| 2 | La boda de mi mejor amiga
| 3 | Alta Fidelidad
| 4 | + El diario de Bridget Jones
| 5 | + El diario de Noa
| 6 | + Barbie
+-----+

sesion3-practica on ✘ master [!]
> git add peliculas.txt
```

```
sesion3-practica on 🌐 master [+]
> lg

+ git commit -m Added 3 new movies -m Closes #2

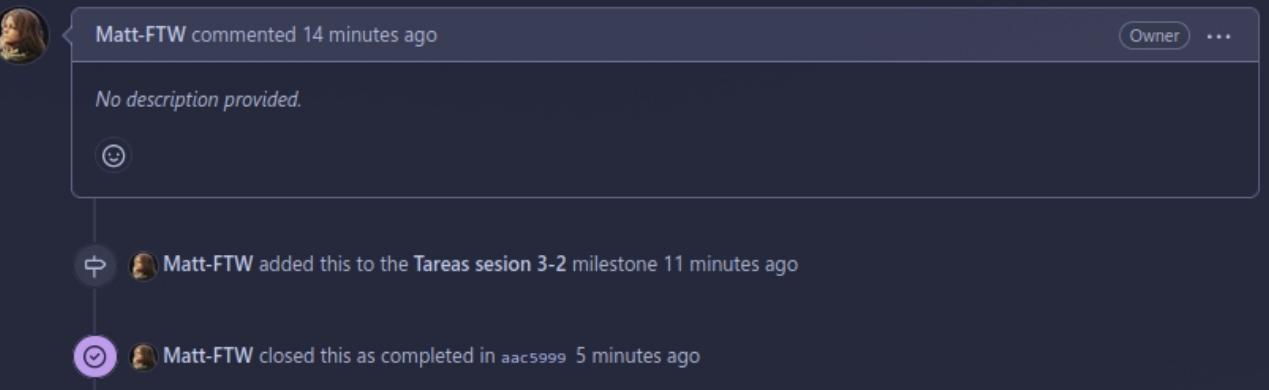
[master aac5999] Added 3 new movies
 1 file changed, 3 insertions(+)

Press enter to return to lazygit

sesion3-practica on 🌐 master [↑] took 49s
> git push
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 24 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 1.00 KiB | 1.00 MiB/s, listo.
Total 3 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Matt-FTW/sesion3-practica.git
  587b243..aac5999  master -> master
```

Añadir 3 películas #2

 Closed Matt-FTW opened this issue 14 minutes ago · 0 comments



Matt-FTW commented 14 minutes ago

No description provided.

Matt-FTW added this to the Tareas sesion 3-2 milestone 11 minutes ago

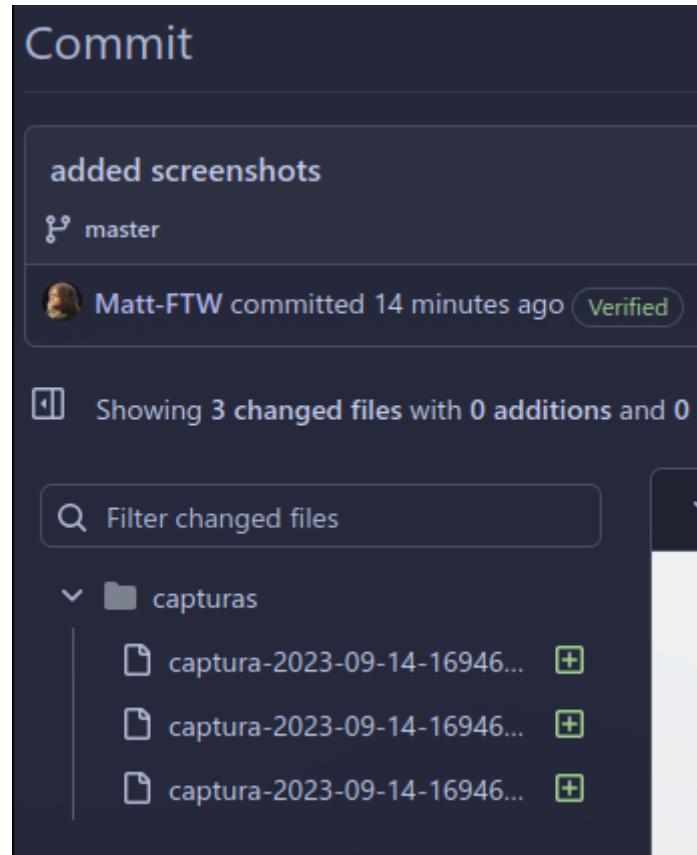
Matt-FTW closed this as completed in aac5999 5 minutes ago

Solo te muestro el commit de las películas para no tener que llenar tanto el documento.

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

Las capturas están arriba

6. 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.



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

