

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.

```
Microsoft Windows [Versión 10.0.22631.4169]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\fjpvi>git --version
git version 2.46.2.windows.1

C:\Users\fjpvi>
```

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

```
C:\Users\fjpvi>git config --global user.name "Nombre y apellido"
```

```
C:\Users\fjpvi>git config --global user.email CORREO@ELECTRONICO
```

```
C:\Users\fjpvi>git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"
C:\Users\fjpvi>
```

```
C:\Users\fjpvi>git config --global user.name
Francisco Pomata

C:\Users\fjpvi>git config --global user.email
fjpvinformatica@gmail.com

C:\Users\fjpvi>
```

```
C:\CURSO-GITHUB\S1R1>GIT CONFIG --LIST
git: 'CONFIG' is not a git command. See 'git --help'.

C:\CURSO-GITHUB\S1R1>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Francisco Pomata
user.email=fjpvinformatica@gmail.com
core.editor='C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin
credential.helper=wincred

C:\CURSO-GITHUB\S1R1>
```

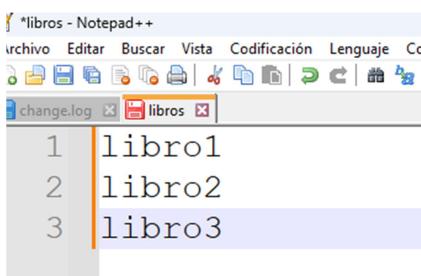
3. Crea una carpeta denominada S1R1. Realiza las siguientes acciones en ella:

```
C:\Símbolo del sistema X + ▾
C:\CURSO-GITHUB\S1R1>|
```

a) Crea un repositorio Git.

```
C:\CURSO-GITHUB\S1R1>git init
Initialized empty Git repository in C:/CURSO-GITHUB/S1R1/.git/
C:\CURSO-GITHUB\S1R1>
```

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

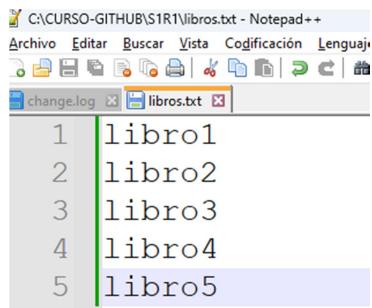


c) Haz un primer commit.

```
C:\CURSO-GITHUB\S1R1>git commit -m "Primera version"
[master (root-commit) afb77cf] Primera version
 1 file changed, 3 insertions(+)
 create mode 100644 libros.txt

C:\CURSO-GITHUB\S1R1>
```

d) Añade dos libros al archivo libros.txt.



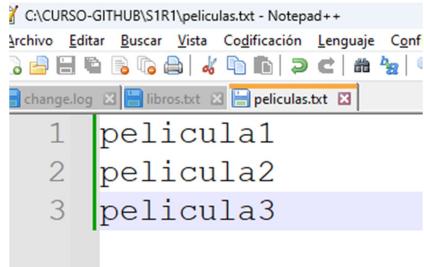
2 FUNDAMENTOS DE GIT I

e) Haz un segundo commit.

```
C:\CURSO-GITHUB\S1R1>git commit -m "Segunda version"
[master 56ddcae] Segunda version
 1 file changed, 3 insertions(+), 1 deletion(-)

C:\CURSO-GITHUB\S1R1>
```

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



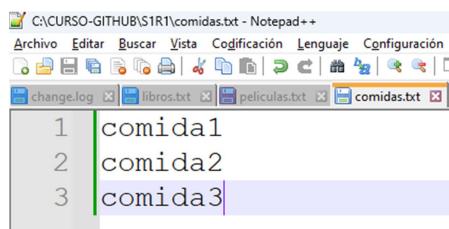
g) Haz una captura de pantalla del comando git status.

```
C:\CURSO-GITHUB\S1R1>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    peliculas.txt

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

C:\CURSO-GITHUB\S1R1>
```

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



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

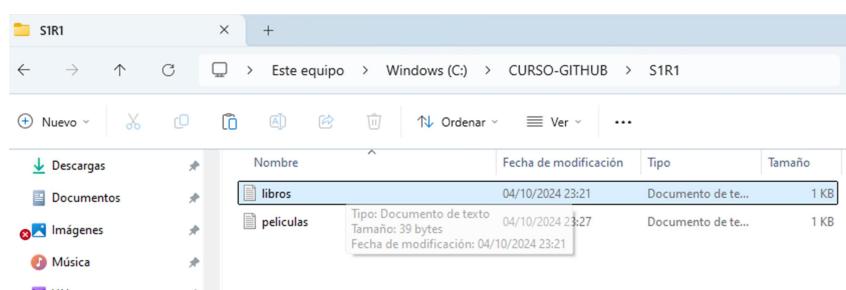
```
C:\CURSO-GITHUB\S1R1>git add .

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

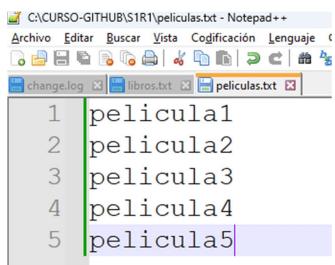
C:\CURSO-GITHUB\S1R1>git commit -m "peliculas.txt y comidas.txt"
[master b347e68] peliculas.txt y comidas.txt
 2 files changed, 6 insertions(+)
 create mode 100644 comidas.txt
 create mode 100644 peliculas.txt

C:\CURSO-GITHUB\S1R1>
```

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

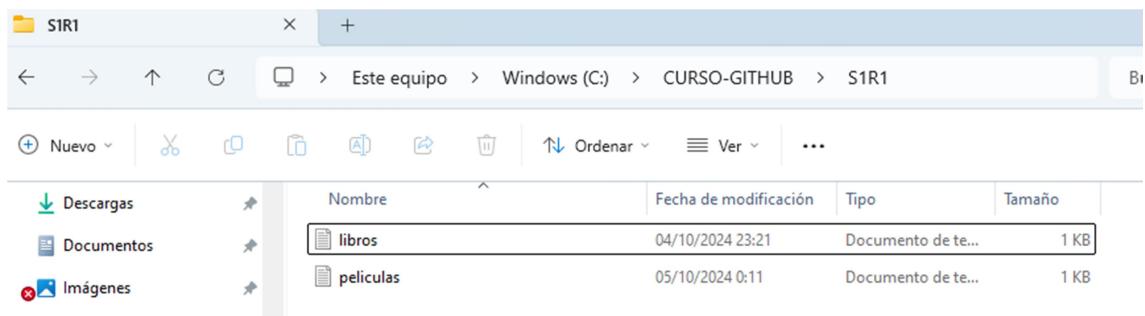


k) Añade dos películas más al archivo peliculas.txt.



```
C:\CURSO-GITHUB\S1R1\peliculas.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje 
[change.log] [libros.txt] [peliculas.txt]
1 pelicula1
2 pelicula2
3 pelicula3
4 pelicula4
5 pelicula5
```

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



m) Añade los cambios al área de preparación.

```
C:\CURSO-GITHUB\S1R1>git add .
```

n) Haz una captura de pantalla del comando git status. Debe indicar que se ha borrado el archivo comidas.txt y que se ha modificado el archivo peliculas.txt.

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

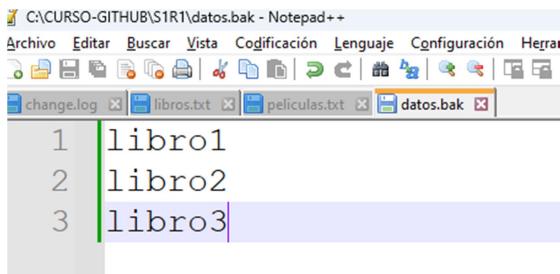
```
C:\CURSO-GITHUB\S1R1>
```

ñ) Haz un cuarto commit.

```
C:\CURSO-GITHUB\S1R1>git commit -m "Cuarto version"
[master 8493350] Cuarto version
 2 files changed, 3 insertions(+), 4 deletions(-)
 delete mode 100644 comidas.txt
```

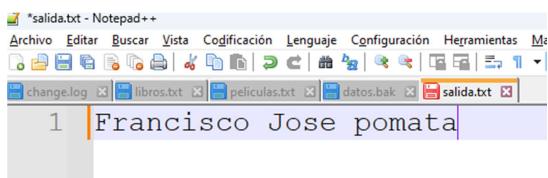
```
C:\CURSO-GITHUB\S1R1>
```

- 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.



```
C:\CURSO-GITHUB\S1R1\datos.bak - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar C
change.log libros.txt peliculas.txt datos.bak salida.txt
1 libro1
2 libro2
3 libro3
```

- 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.

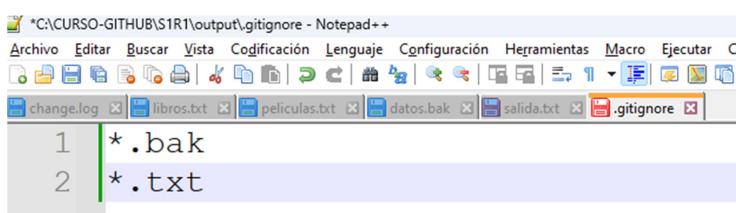


```
*salida.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar C
change.log libros.txt peliculas.txt datos.bak salida.txt
1 Francisco Jose pomata
```

- 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).

```
C:\CURSO-GITHUB\S1R1>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    datos.bak
    output/
nothing added to commit but untracked files present (use "git add" to track)
C:\CURSO-GITHUB\S1R1>
```

- 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.



```
*C:\CURSO-GITHUB\S1R1\output\.gitignore - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar C
change.log libros.txt peliculas.txt datos.bak salida.txt .gitignore
1 *.bak
2 *.txt
```

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.

```
C:\CURSO-GITHUB\S1R1>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

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

C:\CURSO-GITHUB\S1R1>
```

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

```
C:\CURSO-GITHUB\S1R1>git add .gitignore

C:\CURSO-GITHUB\S1R1>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

C:\CURSO-GITHUB\S1R1>git commit -m "archivo gitignore"
[master ba17b4b] archivo gitignore
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore

C:\CURSO-GITHUB\S1R1>
```

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

```
C:\CURSO-GITHUB\S1R1>git log
commit ba17b4b32273f8c2604a4c083b83be2661b91470 (HEAD -> master)
Author: Francisco Pomata <fjpvinformatica@gmail.com>
Date:   Sat Oct 5 16:40:28 2024 +0200

    archivo gitignore

commit 849335052a4d9b2099dbe72e7d366b45152bb0ec
Author: Francisco Pomata <fjpvinformatica@gmail.com>
Date:   Sat Oct 5 00:18:59 2024 +0200

    Cuarta version

commit b347e6854e8c20728a26d630d7dc836794f61528
Author: Francisco Pomata <fjpvinformatica@gmail.com>
Date:   Sat Oct 5 00:07:42 2024 +0200

    peliculas.txt y comedias.txt

commit 56ddcae732547b99d0d2139551bc10f9000ef366
Author: Francisco Pomata <fjpvinformatica@gmail.com>
Date:   Fri Oct 4 23:25:57 2024 +0200

    Segunda version

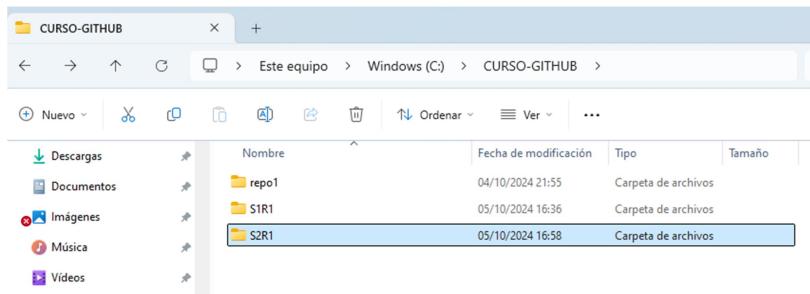
commit afb77cf1b9956844f89c790ff4ce399ed0848f60
Author: Francisco Pomata <fjpvinformatica@gmail.com>
Date:   Fri Oct 4 23:17:55 2024 +0200

    Primera version

C:\CURSO-GITHUB\S1R1>
```

3 FUNDAMENTOS DE GIT II

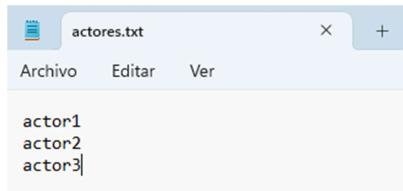
1. Crea una carpeta denominada S2R1. Realiza las siguientes acciones en ella:



a) Crea un repositorio Git.

```
C:\CURSO-GITHUB\S2R1>git init
Initialized empty Git repository in C:/CURSO-GITHUB/S2R1/.git/
C:\CURSO-GITHUB\S2R1>
```

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



c) Haz un primer commit.

```
C:\CURSO-GITHUB\S2R1>git add actores.txt
C:\CURSO-GITHUB\S2R1>git status
On branch master

No commits yet

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

C:\CURSO-GITHUB\S2R1>git commit -m "primera version"
[master (root-commit) ebe8405] primera version
 1 file changed, 3 insertions(+)
 create mode 100644 actores.txt

C:\CURSO-GITHUB\S2R1>
```

d) Crea una rama denominada test.

```
C:\CURSO-GITHUB\repo1>git branch test
C:\CURSO-GITHUB\repo1>git branch
* master
  test

C:\CURSO-GITHUB\repo1>
```

e) Cambia a la rama test

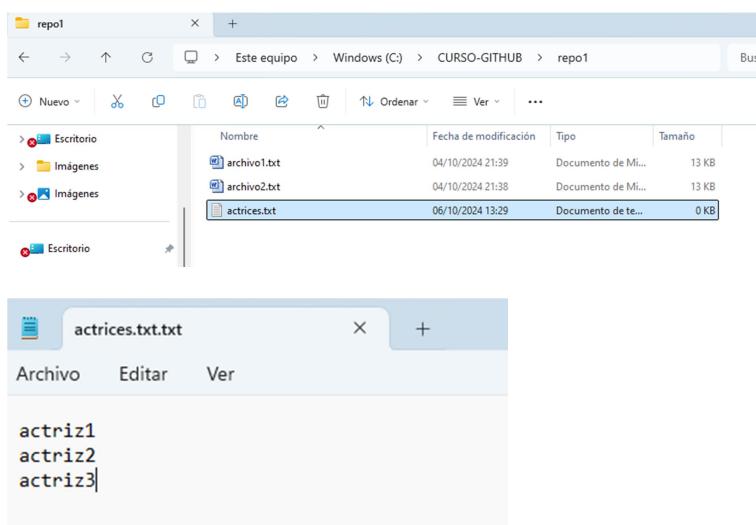
```
C:\CURSO-GITHUB\repo1>git checkout test
D      ~$chivo1.txt.docx
D      ~$chivo2.txt.docx
Switched to branch 'test'

C:\CURSO-GITHUB\repo1>
```

```
C:\CURSO-GITHUB\repo1>git branch
  master
* test
```

```
C:\CURSO-GITHUB\repo1>
```

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



```
C:\CURSO-GITHUB\repo1>git add .

C:\CURSO-GITHUB\repo1>git status
On branch test
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   actrices.txt.txt
    deleted:   ~$chivo1.txt.docx
    deleted:   ~$chivo2.txt.docx

C:\CURSO-GITHUB\repo1>
```

```
C:\CURSO-GITHUB\repo1>git commit -m "Añadí tres actrices"
[test 4b94e5c] Añadí tres actrices
3 files changed, 3 insertions(+)
create mode 100644 actrices.txt.txt
delete mode 100644 ~$chivo1.txt.docx
delete mode 100644 ~$chivo2.txt.docx
```

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

```
C:\CURSO-GITHUB\repo1>git log --graph --all
* commit 4b94e5c5e32c4c7c10296d2ecab3a3858e0f1773 [HEAD -> test]
| Author: Francisco Pomata <fjpvinformatica@gmail.com>
| Date:   Sun Oct 6 13:33:46 2024 +0200
|
|     Añadí tres actrices
|
* commit a0c95eda70dbb27680d731d38be846a59671effe [master]
| Author: Francisco Pomata <fjpvinformatica@gmail.com>
| Date:   Fri Oct 4 21:41:48 2024 +0200
|
|     Nueva lista de ferreteria
|
* commit 6a609ecab93eb37e58ed8f5ca982e478119b55a3
| Author: Francisco Pomata <fjpvinformatica@gmail.com>
| Date:   Fri Oct 4 21:33:07 2024 +0200
|
|     pepinos y peras
|
* commit 38ae759475e76226328afee3a81c4b814b266690
| Author: Francisco Pomata <fjpvinformatica@gmail.com>
| Date:   Fri Oct 4 21:20:25 2024 +0200
|
|     Avena pero no pepinos
|
* commit f382f37a23b3e719f308336ecec54602c85dea0c
| Author: Francisco Pomata <fjpvinformatica@gmail.com>
| Date:   Fri Oct 4 21:08:32 2024 +0200
|
|     Segunda version
|
* commit 8fa38ec6421485d37671a81c9bbb016da6eb4b01
| Author: Francisco Pomata <fjpvinformatica@gmail.com>
| Date:   Fri Oct 4 20:56:55 2024 +0200
|
|     Primera version
C:\CURSO-GITHUB\repo1>
```

h) Cambia a la rama master.

```
C:\CURSO-GITHUB\repo1>git checkout master
Switched to branch 'master'

C:\CURSO-GITHUB\repo1>
```

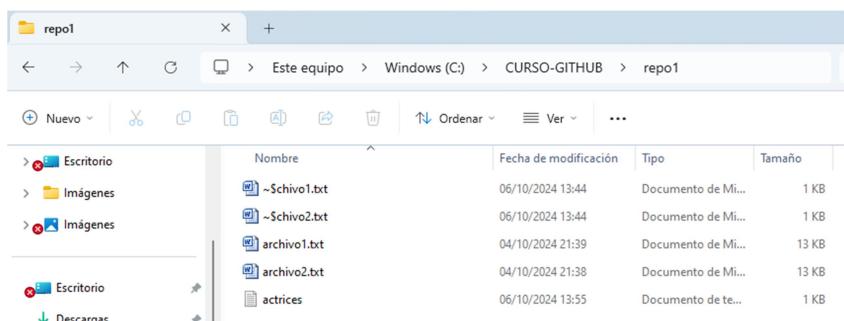
- i) Incorpora los cambios de la rama test a la rama master. Haz una captura de pantalla de los comandos que has utilizado y de su resultado.

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

```
Símbolo del sistema
C:\CURSO-GITHUB\repo1>git branch test2
C:\CURSO-GITHUB\repo1>git branch
* master
  test
  test2

C:\CURSO-GITHUB\repo1>
```

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



```
C:\CURSO-GITHUB\repo1>git commit -m "En master fichero actrices"
[master b6c34aa] En master fichero actrices
 1 file changed, 4 insertions(+)
 create mode 100644 actrices.txt

C:\CURSO-GITHUB\repo1>
```

```
C:\CURSO-GITHUB\repo1>git add .

C:\CURSO-GITHUB\repo1>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   actrices.txt

C:\CURSO-GITHUB\repo1>
```

l) Cambia a la rama test2

```
C:\CURSO-GITHUB\repo1>git checkout test2
Switched to branch 'test2'

C:\CURSO-GITHUB\repo1>
```

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



```
C:\CURSO-GITHUB\repo1>git add .

C:\CURSO-GITHUB\repo1>git status
On branch test2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   actrices.txt

C:\CURSO-GITHUB\repo1>
```

```
C:\CURSO-GITHUB\repo1>git commit -m "En rama test2 + una actriz"
[test2 2a756f7] En rama test2 + una actriz
 1 file changed, 5 insertions(+)
 create mode 100644 actrices.txt

C:\CURSO-GITHUB\repo1>
```

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.

```
Símbolo del sistema - git log × + ▾
* commit 2a756f734a04f12da213acded3b34d2cf6364e50 (HEAD -> test2)
  Author: Francisco Pomata <fjpvinformatica@gmail.com>
  Date:   Sun Oct 6 14:04:37 2024 +0200

    En rama test2 + una actriz

* commit b6c34aa100e18301e0759e8b42f7d0f344fc98f0 (master)
  Author: Francisco Pomata <fjpvinformatica@gmail.com>
  Date:   Sun Oct 6 13:58:36 2024 +0200

    En master fichero actrices
```

ñ) Cambia a la rama master

```
C:\CURSO-GITHUB\repo1>git checkout master
Switched to branch 'master'

C:\CURSO-GITHUB\repo1>
```

o) ¿Se produce un conflicto? De ser así realiza una captura del comando git status.

```
C:\CURSO-GITHUB\repo1>git commit -m "paso de datos a master"
On branch master
nothing to commit, working tree clean

C:\CURSO-GITHUB\repo1>git status
On branch master
nothing to commit, working tree clean

C:\CURSO-GITHUB\repo1>
```

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

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.

4 FUNDAMENTOS DE GIT II

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.

```
C:\CURSO-GITHUB>md S2R2-REMOTO  
C:\CURSO-GITHUB>CD S2R2-REMOTO  
El sistema no puede encontrar la ruta especificada.  
C:\CURSO-GITHUB>CD S2R2-REMOTO  
C:\CURSO-GITHUB\S2R2-REMOTO>
```

```
C:\CURSO-GITHUB\S2R2-REMOTO>git init --bare  
Initialized empty Git repository in C:/CURSO-GITHUB/S2R2-REMOTO/  
C:\CURSO-GITHUB\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:

```
C:\CURSO-GITHUB>git clone S2R2-REMOTO/ S2R2  
Cloning into 'S2R2'...  
warning: You appear to have cloned an empty repository.  
done.  
C:\CURSO-GITHUB>
```

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

```
C:\CURSO-GITHUB>cd s2r2  
C:\CURSO-GITHUB\S2R2>dir  
El volumen de la unidad C es Windows  
El n mero de serie del volumen es: 38AB-DF43  
  
Directorio de C:\CURSO-GITHUB\S2R2  
  
06/10/2024 17:16    <DIR>      .  
06/10/2024 17:14    <DIR>      ..  
06/10/2024 17:16              31 DIRECTORES.TXT.txt  
                  1 archivos          31 bytes  
                  2 dirs   515.055.104.000 bytes libres  
C:\CURSO-GITHUB\S2R2>
```



actrices.txt DIRECTORES.TXT.txt

Archivo Editar Ver

director1
director2
director3

b) Haz un commit.

```
C:\CURSO-GITHUB\S2R2>git status
On branch master

No commits yet

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

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

C:\CURSO-GITHUB\S2R2>git add directores.txt

C:\CURSO-GITHUB\S2R2>
```

```
C:\CURSO-GITHUB\S2R2>git commit -m "fichero directores"
[master (root-commit) fa86926] fichero directores
 1 file changed, 3 insertions(+)
 create mode 100644 directores.txt

C:\CURSO-GITHUB\S2R2>
```

```
C:\CURSO-GITHUB\S2R2>git log
commit fa8692650afa3b2efc68b0fc0ef1dae09a689105 (HEAD -> master)
Author: Francisco Pomata <fjpvinformatica@gmail.com>
Date:   Sun Oct 6 17:31:02 2024 +0200

  fichero directores

C:\CURSO-GITHUB\S2R2>
```

```
C:\CURSO-GITHUB\S2R2>git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean

C:\CURSO-GITHUB\S2R2>
```

```
C:\CURSO-GITHUB\S2R2>git remote
origin
```

```
C:\CURSO-GITHUB\S2R2>git remote -v
origin  C:/CURSO-GITHUB/S2R2-REMOTO/ (fetch)
origin  C:/CURSO-GITHUB/S2R2-REMOTO/ (push)

C:\CURSO-GITHUB\S2R2>
```

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

```
C:\CURSO-GITHUB\S2R2>git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 245 bytes | 245.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To C:/CURSO-GITHUB/S2R2-REMOTO/
 * [new branch]      master -> master

C:\CURSO-GITHUB\S2R2>
```

```
C:\CURSO-GITHUB\S2R2>git log --all
commit fa8692650afa3b2efc68b0fc0ef1dae09a689105 (HEAD -> master, origin/master)
Author: Francisco Pomata <fjpvinformatica@gmail.com>
Date:   Sun Oct 6 17:31:02 2024 +0200

    fichero directores

C:\CURSO-GITHUB\S2R2>
```

d) Crea una rama denominada version1.

```
Símbolo del sistema      + | v
C:\CURSO-GITHUB\S2R2>git branch version1
C:\CURSO-GITHUB\S2R2>git branch
* master
  version1

C:\CURSO-GITHUB\S2R2>
```

e) Cambia a la rama version1.

```
C:\CURSO-GITHUB\S2R2>git checkout version1
Switched to branch 'version1'

C:\CURSO-GITHUB\S2R2>
```

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.



```
C:\CURSO-GITHUB\S2R2>git add directores.txt  
C:\CURSO-GITHUB\S2R2>git status  
On branch version1  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    modified:   directores.txt  
  
C:\CURSO-GITHUB\S2R2>
```

```
C:\CURSO-GITHUB\S2R2>git commit -m "añado dos directores al fichero directores"  
[version1 550535c] añado dos directores al fichero directores  
 1 file changed, 3 insertions(+), 1 deletion(-)  
  
C:\CURSO-GITHUB\S2R2>
```

```
C:\CURSO-GITHUB\S2R2>git log  
commit 550535c7022497d9451865b0082c7e146c59cbae (HEAD -> version1)  
Author: Francisco Pomata <fjpvinformatica@gmail.com>  
Date:   Sun Oct 6 17:51:21 2024 +0200  
  
  añado dos directores al fichero directores  
  
commit fa8692650afa3b2efc68b0fc0ef1dae09a689105 (origin/master, master)  
Author: Francisco Pomata <fjpvinformatica@gmail.com>  
Date:   Sun Oct 6 17:31:02 2024 +0200  
  
  fichero directores  
  
C:\CURSO-GITHUB\S2R2>
```

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.

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

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.

5 GITHUB

d) Cambia a la rama master.

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

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

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.

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.

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

4. GitHub

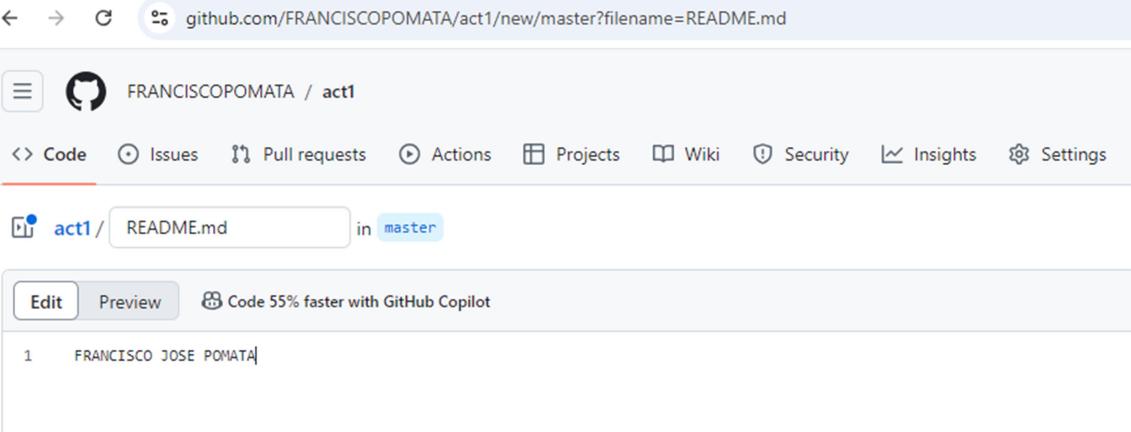
1. Crea una cuenta en GitHub

2. Añade tu dirección de correo de educación

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

4. Haz un fork del repositorio localizado en la siguiente url: <https://github.com/curso-github-cefire/sesion3-practica>. A partir de este momento todas las tareas que se indican se deben realizar en tu repositorio (el que has clonado mediante el fork).

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



The screenshot shows a GitHub repository named 'act1' belonging to user 'FRANCISCOPOMATA'. The 'Code' tab is selected. A search bar at the top shows 'act1 / README.md in master'. Below the search bar, there are buttons for 'Edit' and 'Preview', with a note 'Code 55% faster with GitHub Copilot'. The main area displays a single line of code: '1 FRANCISCO JOSE POMATA|'. This represents the first commit made to the repository.

- 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.

6 ENTREGA DE LA TAREA

- 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.

- f) Incluye las capturas de pantalla en el repositorio dentro de la carpeta capturas. Añádelas también al repositorio de manera que queden guardadas en tu repositorio en GitHub.

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

5. Entrega de la tarea

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

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