

Práctica - Uso de Git y GitHub

Índice

1. Preparación	2
2. Fundamentos de Git I	2
3. Fundamentos de Git II	4
4. GitHub	6
5. Entrega de la tarea	7

1. Preparación

Realiza las tareas que se indican a continuación. Incluye las **capturas de pantalla** que se pidan en un documento tipo **LibreOffice** o **Word**.

Cuando se pida realizar un *commit* recuerda que previamente hay que añadir los archivos al área de preparación si no se ha indicado antes en las instrucciones. En esos casos, un *commit* significa ejecutar los comandos `git add` `git commit`.

¡IMPORTANTE! No utilices el bloc de notas de Windows para editar los archivos de texto de las tareas. Utiliza en su lugar un editor específico. Algunas sugerencias son:

- Visual Studio Code
- Atom
- Brackets
- Sublime Text
- Notepad ++

El bloc de notas de Windows utiliza una codificación de caracteres específica de Windows y además incluye la extensión `.txt` por defecto al final de los nombres de archivo. Por tanto, no es una buena elección para trabajar.

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`.

```
PS C:\WINDOWS\system32> git --version
git version 2.52.0.windows.1
```

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

```
PS C:\WINDOWS\system32> git config --global user.name "David Albar Martínez"
PS C:\WINDOWS\system32> git config --global user.email "daibar25@gmail.com"
PS C:\WINDOWS\system32> git config --global core.editor "C:\Users\Dejvi\AppData\Local\Programs\Microsoft VS Code\ -multiInst -notabbar -nosession -noPlugin"
```

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

- a) Crea un repositorio Git.

```
PS C:\Users\Deyvi\Desktop\PPS> mkdir S1R1

Directorio: C:\Users\Deyvi\Desktop\PPS

Mode                LastWriteTime         Length Name
----                -
d-----          27/11/2025   16:16         S1R1
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git add libros.txt
```

- c) Haz un primer *commit*.

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git commit -m "Primer Commit: Lista de Libros"
[master (root-commit) 0e17168] Primer Commit: Lista de Libros
 1 file changed, 3 insertions(+)
 create mode 100644 libros.txt
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git add libros.txt
```

- e) Haz un segundo *commit*.

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git commit -m "Segundo Commit: Lista de Libros Actualizada"
[master d4cdedd] Segundo Commit: Lista de Libros Actualizada
 1 file changed, 3 insertions(+), 1 deletion(-)
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git add peliculas.txt
```

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

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

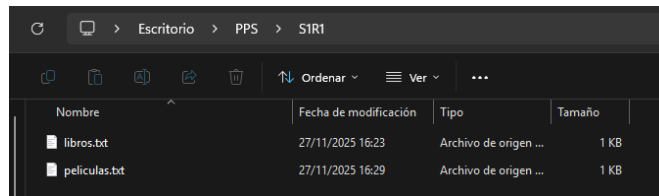
- h) Crea un fichero denominado comidas.txt. Añade tres nombres de comidas a dicho archivo.

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git add comidas.txt
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git commit -m "Tercer Commit: Lista de Peliculas y Comidas"
[master a38a2a9] Tercer Commit: Lista de Peliculas y Comidas
1 file changed, 3 insertions(+)
create mode 100644 peliculas.txt
```

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



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

```
C: > Users > Deyvi > Desktop > PPS > S1R1 > ⚙️ peliculas.txt
1 IT
2 Superman
3 Codigo Enigma
4 Spiderman Far From Home
5 Avatar
```

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

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

- m) Añade los cambios al área de preparación.
- n) Haz una captura de pantalla del comando git status. Debe indicar que se ha borrado el archivo comidas.txt y que se ha modificado el archivo peliculas.txt.

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> 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*.

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git commit -m "Cuarto Commit: Lista actualizada de Peliculas y Comidas eliminada"
```

- 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: > Users > Deyvi > Desktop > PPS > S1R1 > ⚙️ datos.bak
1 The Witcher
2 Metro 2033
3 El Corredor del Laberinto
```

- 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.**
- 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).

```
PS C:\Users\Deyvi\Desktop\PPS\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)
```

- 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.

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> Set-Content .gitignore "*.bak"r\noutput/"
```

- 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.

```
PS C:\Users\Deyvi\Desktop\PPS\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)
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git commit -m "Ultimo Commit: Se añade el ignore"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S1R1> git log
commit c2a87e9df4bfb0594fd3d39581f9ef2ebbb3757 (HEAD --> master)
Author: David Aibar Martinez <daibar25@gmail.com>
Date: Thu Nov 27 22:21:58 2025 +0100

    Ultimo Commit: Se añade el ignore

commit 7eadbeb311526bc17ca55e06d239d09294f30c8f
Author: David Aibar Martinez <daibar25@gmail.com>
Date: Thu Nov 27 16:40:44 2025 +0100

    Cuarto Commit: Lista actualizada de Peliculas y Comidas eliminada

commit 18b0c8487a3bcff671be1d8666ab6b876aa454b8
Author: David Aibar Martinez <daibar25@gmail.com>
Date: Thu Nov 27 16:38:25 2025 +0100

    Tercer Commit: Lista de Peliculas y Comidas

commit a38a2a9f89267682146ac25b07c4e7bb065280f3
Author: David Aibar Martinez <daibar25@gmail.com>
Date: Thu Nov 27 16:32:28 2025 +0100
```

3. Fundamentos de Git II

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

- a) Crea un repositorio Git.

```
PS C:\Users\Deyvi\Desktop\PPS> mkdir S2R1

Directorio: C:\Users\Deyvi\Desktop\PPS

Mode                LastWriteTime         Length Name
----                -
d-----          28/11/2025   17:05             S2R1

PS C:\Users\Deyvi\Desktop\PPS> cd S2R1
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git init
Initialized empty Git repository in C:/Users/Deyvi/Desktop/PPS/S2R1/.git/
```

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

```
actores.txt X  Practica6.md ●

C: > Users > Deyvi > Desktop > PPS > S2R1 > actores.txt
1  Nicolas Hoult
2  Aaron Pierre
3  David Harbour
```

- c) Haz un primer *commit*.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git add actores.txt
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git commit -m "Primer Commit: Actores añadidos"
[master (root-commit) a14f7b7] Primer Commit: Actores añadidos
1 file changed, 3 insertions(+)
create mode 100644 actores.txt
```

- d) Crea una rama denominada test.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git branch test
```

- e) Cambia a la rama test

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> 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.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> echo "Julia Roberts\nAngelina Jolie\nScarlett Johansson" > actrices.txt
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git log --graph --all --oneline
* 2a3f18d (HEAD -> test) Actrices añadidas en rama test
* a14f7b7 (master) Primer Commit: Actores añadidos
```

- h) Cambia a la rama master.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git checkout master
M      actores.txt
Switched to branch 'master'
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git checkout master
M      actores.txt
Switched to branch 'master'
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git merge test
Updating a14f7b7..2a3f18d
Fast-forward
 actores.txt | Bin 0 -> 100 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 actores.txt
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git branch test2
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> Add-Content actores.txt "Isabela Merced"
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git commit -m "Modificacion master: Isabela"
[master 1a100f3] Modificacion master: Isabela
```

- l) Cambia a la rama test2

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git checkout test2
M      actores.txt
Switched to branch 'test2'
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> Add-Content actores.txt "Sadie Sink"
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git add actores.txt
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git commit -m "Modificacion test2: Sadie"
[test2 89e3319] Modificacion test2: Sadie
1 file changed, 0 insertions(+), 0 deletions(-)
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git log --graph --all --oneline
* 89e3319 (HEAD -> test2) Modificacion test2: Sadie
| * 1a100f3 (master) Modificacion master: Isabela
|/
* 2a3f18d (test) Actrices a adidas en rama test
* a14f7b7 Primer Commit: Actores a adidos
```

-  ) Cambia a la rama `master`

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git checkout master
M      actores.txt
Switched to branch 'master'
```

- o) Incorpora los cambios de la rama `test2` a la rama `master`.  Se produce un conflicto? De ser as  realiza una captura del comando `git status`.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   actrices.txt
```

- 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.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R1> git log --graph --all --oneline
* be2fa90 (HEAD -> master) Conflicto Resuelto: a adidas ambas
| \
| * 89e3319 (test2) Modificacion test2: Sadie
| * 1a100f3 Modificacion master: Isabela
|/
* 2a3f18d (test) Actrices a adidas en rama test
* a14f7b7 Primer Commit: Actores a adidos
```

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

```
PS C:\Users\Deyvi\Desktop\PPS> mkdir S2R2-remoto

Directorio: C:\Users\Deyvi\Desktop\PPS

Mode                LastWriteTime         Length Name
----                -
d-----           28/11/2025    17:43             S2R2-remoto

PS C:\Users\Deyvi\Desktop\PPS> cd S2R2-remoto
PS C:\Users\Deyvi\Desktop\PPS\S2R2-remoto> git init --bare
```


3. Clona el repositorio S2R2-remoto en una carpeta denominada S2R2. Adjunta captura de pantalla del resultado del comando de clonado. A conti- nuación realiza las siguientes acciones en el repositorio S2R2:

```
PS C:\Users\Deyvi\Desktop\PPS> git clone S2R2-remoto S2R2
Cloning into 'S2R2'...
warning: You appear to have cloned an empty repository.
done.
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R2> echo "Steven Spielberg'nJames Gunn'nAndy Muschietti" > directores.
txt
```

- b) Haz un *commit*.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R2> git commit -m "Añadidos Directores"
[master (root-commit) 3cc23e3] Añadidos Directores
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 directores.txt
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R2> git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To C:/Users/Deyvi/Desktop/PPS/S2R2-remoto
* [new branch]      master -> master
```

- d) Crea una rama denominada version1.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R2> git branch version1
```

- e) Cambia a la rama version1.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R2> git checkout version1
Switched to branch 'version1'
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R2> Add-Content directores.txt "'nGuillermo Del Toro'nTim Burton"
PS C:\Users\Deyvi\Desktop\PPS\S2R2> git add directores.txt
PS C:\Users\Deyvi\Desktop\PPS\S2R2> git commit -m "Mas directores"
[version1 3156212] Mas directores
1 file changed, 0 insertions(+), 0 deletions(-)
```

- f) Realiza un *push* de la rama al repositorio remoto de manera que **quede asociada a la rama remota del mismo nombre**. Adjunta captura de pantalla del resultado.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R2> git push -u origin version1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 366 bytes | 366.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To C:/Users/Deyvi/Desktop/PPS/S2R2-remoto
* [new branch]      version1 -> version1
branch 'version1' set up to track 'origin/version1'.
```

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.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R3> type directores.txt
Steven Spielberg'nJames Gunn'nAndy Muschietti
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R3> type directores.txt
Steven Spielberg'nJames Gunn'nAndy Muschietti
'nGuillermo Del Toro'nTim Burton
```

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.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R3> type directores.txt
Steven Spielberg'nJames Gunn'nAndy Muschietti
'nGuillermo Del Toro'nTim Burton
```

- d) Cambia a la rama master.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R3> git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R3> git merge version1
Updating 3cc23e3..3156212
Fast-forward
 directores.txt | Bin 96 -> 164 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Deyvi\Desktop\PPS\S2R3> git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To C:\Users\Deyvi\Desktop\PPS\S2R2-remoto
 3cc23e3..3156212 master -> master
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\S2R3> git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To C:\Users\Deyvi\Desktop\PPS\S2R2-remoto
 3cc23e3..3156212 master -> master
```

5. Vuelve de nuevo a la carpeta S2R2y 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.

```
PS C:\Users\Deyvi\Desktop\PPS\S2R2> git fetch
From C:/Users/Deyvi/Desktop/PPS/S2R2-remoto
3cc23e3..3156212 master -> origin/master
```

- b) Actualiza la rama masterlocal con el contenido de la rama master del repositorio remoto. Adjunta captura de pantalla del resultado del comando utilizado.

```
PS C:\Users\Deyvi\Desktop\PPS\s2r2> git pull origin master
From C:/Users/Deyvi/Desktop/PPS/S2R2-remoto
* branch      master      -> FETCH_HEAD
Already up to date.
```

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

```
PS C:\Users\Deyvi\Desktop\PPS\s2r2> type directores.txt
Steven Spielberg'nJames Gunn'nAndy Muschietti
'nGuillermo Del Toro'nTim Burton
```

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*).

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner *

Repository name *

Daibar25

/

sesion3-practica

✓

sesion3-practica is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

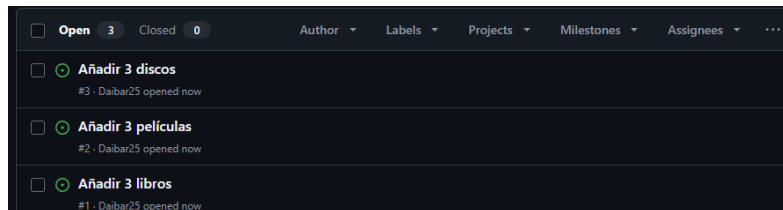
```
PS C:\Users\Deyvi\Desktop\PPS> git clone https://github.com/Daibar25/sesion3-practica.git
Cloning into 'sesion3-practica'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 6 (from 1)
Receiving objects: 100% (10/10), done.
```

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

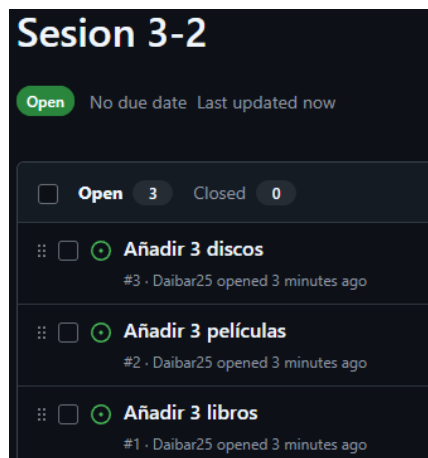
```
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git add README.md
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git commit -m "Añido mi nombre al README"
[master 36ee070] Añido mi nombre al README
1 file changed, 1 insertion(+)
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git push
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 331 bytes | 331.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Daibar25/sesion3-practica.git
a692f8f..36ee070 master -> master
```

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

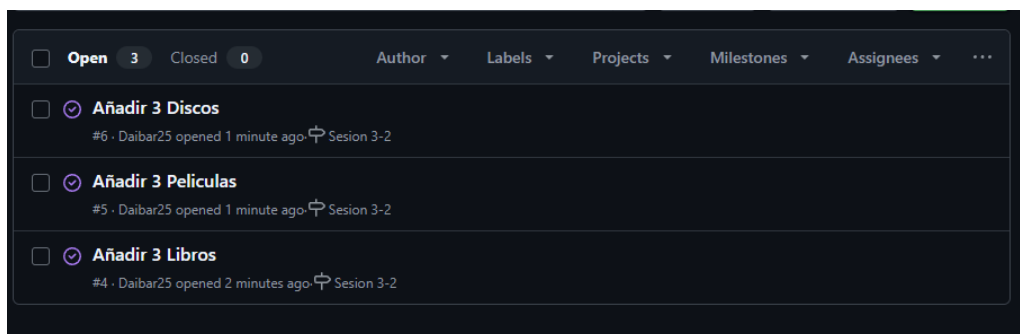
- Añadir 3 libros
- Añadir 3 películas
- Añadir 3 discos



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.

```
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git add libros.txt
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git commit -m "Libros añadidos. Close #4"
[master 78ddb07] Libros añadidos. Close #4
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git add peliculas.txt
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git commit -m "Peliculas añadidas. Close #5"
[master cbc8852] Peliculas añadidas. Close #5
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git add discos.txt
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git commit -m "Discos añadidos. Close #6"
[master 3a53906] Discos añadidos. Close #6
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Deyvi\Desktop\PPS\sesion3-practica> git push --force
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 16 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1.05 KiB | 1.05 MiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Daibar25/sesion3-practica.git
+ 847b8ba...3a53906 master -> master (forced update)
```

- 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.