

Práctica 1 – Uso de Git y Git Hub

1.- Fundamentos Git I:

1.-Captura de pantalla en la que aparezca el resultado de la ejecución del comando git --version.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop
$ git --version
git version 2.23.0.windows.1
```

2.-Captura de pantalla con el resultado de la ejecución de los comandos de configuración.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
credential.helper=manager
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge --skip -- %f
filter.lfs.process=git-lfs filter-process --skip
filter.lfs.required=true
user.name=Manuel Soler Vilar
user.email=manuelsovil@gmail.com
credential.helper=wincred
core.editor=notepad
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop
$ mkdir S1R1
```

a) Crea un repositorio Git.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1
$ git init
Initialized empty Git repository in C:/Users/Manuel/Desktop/S1R1/.git/
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ nano libros.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ cat libros.txt
Los Tres Mosqueteros
El Conde de Montecristo
La Historia Interminable
```

Práctica 1 – Uso de Git y Git Hub

c) Haz un primer Commit.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git commit -m "Primer Commit"
[master (root-commit) 8527bc9] Primer Commit
1 file changed, 3 insertions(+)
create mode 100644 libros.txt
```

d) Añade dos libros al archivo libros.txt

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ nano libros.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git diff
warning: LF will be replaced by CRLF in libros.txt.
The file will have its original line endings in your working directory
diff --git a/libros.txt b/libros.txt
index 77c39c2..c2054b8 100644
--- a/libros.txt
+++ b/libros.txt
@@ -1,3 +1,5 @@
 Los Tres Mosqueteros
 El Conde de Montecristo
 La Historia Interminable
+El Quijote
+La Isla del Tesoro
```

e) Haz un segundo Commit

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git commit -m "Segundo Commit"
[master 403778c] Segundo Commit
1 file changed, 2 insertions(+)
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ nano peliculas.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ cat peliculas.txt
Los Inmortales
Tron
La Guerra de las Galaxias
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    peliculas.txt

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

Práctica 1 – Uso de Git y Git Hub

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

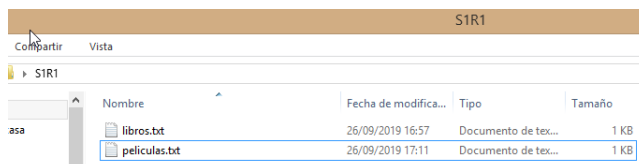
```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ nano comidas.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ cat comidas.txt
Macarrones
Lentejas
Paella
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git commit -m "Tercer Commit Peliculas y Comidas"
[master 05c20a3] Tercer Commit Peliculas y Comidas
2 files changed, 6 insertions(+)
create mode 100644 comidas.txt
create mode 100644 peliculas.txt
```

j) Elimina el archivo comidas.txt



Nombre	Fecha de modifica...	Tipo	Tamaño
libros.txt	26/09/2019 16:57	Documento de tex...	1 KB
peliculas.txt	26/09/2019 17:11	Documento de tex...	1 KB

k) Añade dos películas más al archivo peliculas.txt

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ nano peliculas.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git diff
warning: LF will be replaced by CRLF in peliculas.txt.
The file will have its original line endings in your working directory
diff --git a/peliculas.txt b/peliculas.txt
index 99ac5a4..3a6dc8f 100644
--- a/peliculas.txt
+++ b/peliculas.txt
@@ -1,3 +1,5 @@
 Los Inmortales
 Tron
 La Guerra de las Galaxias
+Terminator
+Top Gun
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git status
On branch master
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")
```

m) Añade los cambios al área de preparación.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git add .
warning: LF will be replaced by CRLF in peliculas.txt.
The file will have its original line endings in your working directory
```

Práctica 1 – Uso de Git y Git Hub

n) Haz una captura de pantalla de git status. (borrado comidas.txt – modificado películas)

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   comidas.txt
    modified:  peliculas.txt
```

ñ) Haz un cuarto commit

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git commit -m "Cuarto Commit Archivos Modificados"
[master 38a2320] Cuarto Commit Archivos Modificados
 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.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ nano datos.bak

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ cat datos.bak
Crimen y castigo
Dracula
La Momia
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ mkdir output

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ cd output

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1/output (master)
$ nano salida.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1/output (master)
$ cat salida.txt
Manuel Soler Vilar
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    datos.bak
    output/
```

Práctica 1 – Uso de Git y Git Hub

No he conseguido sacar output/salida.txt más que haciendo un git add, esto tiene un problema y es que cuando quiero hacer el commit como ya los he seleccionado él .gitignore ya no funciona y me toca deseleccionarlos. Supongo que este fallo será por la edición de git que tengo instalada.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git add *
warning: LF will be replaced by CRLF in datos.bak.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in output/salida.txt.
The file will have its original line endings in your working directory

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   datos.bak
        new file:   output/salida.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ nano .gitignore

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ ls
datos.bak  libros.txt  output/  peliculas.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ cat .gitignore
*.bak
output/*

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   datos.bak
        new file:   output/salida.txt

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

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.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ nano .gitignore

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ cat .gitignore
*.bak
output/*
```

s) Haz una nueva captura de pantalla del comando git status. Ahora no deben aparecer los archivos de datos.bak y output/salida.txt como archivos nuevos, sino que en su lugar debe aparecer únicamente el archivo .gitignore.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
```

Práctica 1 – Uso de Git y Git Hub

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git commit -m "Ultimo Commit .gitignore"
[master 05c3393] Ultimo Commit .gitignore
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S1R1 (master)
$ git log
commit 05c33935510039fbbe4c10d7a23f272f7686fe58 (HEAD -> master)
Author: Manuel Soler Vilar <manuelsovil@gmail.com>
Date: Sat Sep 28 12:48:00 2019 +0200

    Ultimo Commit .gitignore

commit 38a232026e00fd6b67b21f95d8c9ca8d73716ed7
Author: Manuel Soler Vilar <manuelsovil@gmail.com>
Date: Sat Sep 28 12:37:49 2019 +0200

    Cuarto Commit Archivos Modificados

commit 05c20a347d6db91c9786e24f09814c679a04ba62
Author: Manuel Soler Vilar <manuelsovil@gmail.com>
Date: Sat Sep 28 12:29:53 2019 +0200

    Tercer Commit Peliculas y Comidas

commit 403778cc66135cdadf16388e0cacdd8a3f3a113f
Author: Manuel Soler Vilar <manuelsovil@gmail.com>
Date: Sat Sep 28 12:24:14 2019 +0200

    Segundo Commit

commit 8527bc90d23f0680d56f2b81797711a254300605
Author: Manuel Soler Vilar <manuelsovil@gmail.com>
Date: Sat Sep 28 12:21:29 2019 +0200

    Primer Commit
```

2.-Fundamentos de Git II

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop
$ mkdir S2R1
```

a) Crea un repositorio Git

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1
$ git init
Initialized empty Git repository in C:/Users/Manuel/Desktop/S2R1/.git/
```

Práctica 1 – Uso de Git y Git Hub

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ nano actores.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ cat actores.txt
Tom Cruise
Mario Moreno Cantinflas
Jose Luis Lopez Vazquez
```

c) Haz un primer commit.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git commit -m "Primer Commit Git2"
[master (root-commit) 5e3f8a4] Primer Commit Git2
1 file changed, 3 insertions(+)
create mode 100644 actores.txt
```

d) Crea una rama denominada test.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git branch test
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test)
$ nano actrices.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test)
$ cat actrices.txt
Amaia Salamanca
Blanca Suarez
Maribel Verdu
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test)
$ git commit -m "Segundo Commit Git2 Actrices"
[test 0c00787] Segundo Commit Git2 Actrices
1 file changed, 3 insertions(+)
create mode 100644 actrices.txt
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test)
$ git log --graph --all
* commit 0c007872e63bcdacaaed3fdf19bb201effc5ea46 (HEAD -> test)
  Author: Manuel Soler Vilar <manuelsolvil@gmail.com>
  Date: Sat Sep 28 13:06:19 2019 +0200

    Segundo Commit Git2 Actrices

* commit 5e3f8a468733829e7779534e70e91baf327cca05 (master)
  Author: Manuel Soler Vilar <manuelsolvil@gmail.com>
  Date: Sat Sep 28 12:59:51 2019 +0200

    Primer Commit Git2
```

Práctica 1 – Uso de Git y Git Hub

g) 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.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test)
$ git checkout master
Switched to branch 'master'

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git merge test
Updating 5e3f8a4..0c00787
Fast-forward
 actrices.txt | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 actrices.txt
```

h) Posiciónate en la rama master.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git branch
* master
  test
```

i) Crea una segunda rama denominada test2.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git branch test2
```

j) En la rama master, añade una actriz al fichero actrices.txt.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ nano actrices.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git diff
warning: LF will be replaced by CRLF in actrices.txt.
The file will have its original line endings in your working directory
diff --git a/actrices.txt b/actrices.txt
index a72f130..e122481 100644
--- a/actrices.txt
+++ b/actrices.txt
@@ -1,3 +1,4 @@
 Amaia Salamanca
 Blanca Suarez
 Maribel Verdu
+Belén Rueda
```

```
└─
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git commit -m "Tercer Commit Git2"
[master fb5fdea] Tercer Commit Git2
 1 file changed, 1 insertion(+)
```

k) En la rama test2, añade una actriz al fichero actrices.txt.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test2)
$ git diff
warning: LF will be replaced by CRLF in actrices.txt.
The file will have its original line endings in your working directory
diff --git a/actrices.txt b/actrices.txt
index a72f130..8d62b67 100644
--- a/actrices.txt
+++ b/actrices.txt
@@ -1,3 +1,4 @@
 Amaia Salamanca
 Blanca Suarez
 Maribel Verdu
+Inma Cuesta
```


Práctica 1 – Uso de Git y Git Hub

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test2)
$ git commit -m "Cuarto Commit Git2"
[test2 f706729] Cuarto Commit Git2
1 file changed, 1 insertion(+)
```

l) 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.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test2)
$ git log --graph --all
* commit f70672952592cda2f4b32348b82055ba0fb5c9f0 (HEAD -> test2)
  Author: Manuel Soler Vilar <manuelsolvil@gmail.com>
  Date: Sat Sep 28 13:24:12 2019 +0200

    Cuarto Commit Git2

* commit fb5fdea0cce12746dddb6ad2b54a26518dcf78bf (master)
  Author: Manuel Soler Vilar <manuelsolvil@gmail.com>
  Date: Sat Sep 28 13:20:26 2019 +0200

    Tercer Commit Git2

* commit 0c007872e63bcdacaed3fdf19bb201effc5ea46 (test)
  Author: Manuel Soler Vilar <manuelsolvil@gmail.com>
  Date: Sat Sep 28 13:06:19 2019 +0200

    Segundo Commit Git2 Actrices

* commit 5e3f8a468733829e7779534e70e91baf327cca05
  Author: Manuel Soler Vilar <manuelsolvil@gmail.com>
  Date: Sat Sep 28 12:59:51 2019 +0200

    Primer Commit Git2
```

m) 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`.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (test2)
$ git checkout master
Switched to branch 'master'

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git merge test2
Auto-merging actrices.txt
CONFLICT (content): Merge conflict in actrices.txt
Automatic merge failed; fix conflicts and then commit the result.

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

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

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

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master|MERGING)
$ nano actrices.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master|MERGING)
$ cat actrices.txt
Amaia Salamanca
Blanca Suarez
Maribel Verdu
Belén Rueda
Inma Cuesta
```

Práctica 1 – Uso de Git y Git Hub

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master|MERGING)
$ git add .

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  modified:   actrices.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master|MERGING)
$ git commit -m "Ultimo Commit Git2 Resolucion Conflicto"
[master 1f0e0ab] Ultimo Commit Git2 Resolucion Conflicto
```

ñ) 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.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R1 (master)
$ git log --graph --all
* commit 1f0e0abbbb4311ea211b947a4f10c4b0f8dc85f (HEAD -> master)
| Merge: fb5fdea f706729
| Author: Manuel Soler Vilar <manuelsovil@gmail.com>
| Date: Sat Sep 28 13:36:38 2019 +0200
|
| Ultimo Commit Git2 Resolucion Conflicto
|
* commit f70672952592cda2f4b32348b82055ba0fb5c9f0 (test2)
| Author: Manuel Soler Vilar <manuelsovil@gmail.com>
| Date: Sat Sep 28 13:24:12 2019 +0200
|
| Cuarto Commit Git2
|
* commit fb5fdea0cce12746dddb6ad2b54a26518dcf78bf
| Author: Manuel Soler Vilar <manuelsovil@gmail.com>
| Date: Sat Sep 28 13:20:26 2019 +0200
|
| Tercer Commit Git2
|
* commit 0c007872e63bcdacaaed3fdf19bb201effc5ea46 (test)
| Author: Manuel Soler Vilar <manuelsovil@gmail.com>
| Date: Sat Sep 28 13:06:19 2019 +0200
|
| Segundo Commit Git2 Actrices
|
* commit 5e3f8a468733829e7779534e70e91baf327cca05
| Author: Manuel Soler Vilar <manuelsovil@gmail.com>
| Date: Sat Sep 28 12:59:51 2019 +0200
|
| Primer Commit Git2
```

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.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2-remoto
$ git init --bare
Initialized empty Git repository in C:/Users/Manuel/Desktop/S2R2-remoto/
```

Práctica 1 – Uso de Git y Git Hub

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop
$ 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.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (master)
$ nano directores.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (master)
$ cat directores.txt
Steven Spielberg
Tim Burton
Juan Antonio Ballona
```

b) Haz un commit.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (master)
$ git commit -m "Primer Commit Git2 Parte 3"
[master (root-commit) 0b33f7c] Primer Commit Git2 Parte 3
1 file changed, 3 insertions(+)
create mode 100644 directores.txt
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 284 bytes | 71.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To C:/Users/Manuel/Desktop/S2R2-remoto
* [new branch]      master -> master
```

d) Crea una rama denominada version1

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (master)
$ git branch version1
```

e) En la rama version1 añade el nombre de dos directores de cine más al archivo directores.txt

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (version1)
$ git diff
warning: LF will be replaced by CRLF in directores.txt.
The file will have its original line endings in your working directory
diff --git a/directores.txt b/directores.txt
index 78ee18e..8e9680f 100644
--- a/directores.txt
+++ b/directores.txt
@@ -1,3 +1,5 @@
 Steven Spielberg
 Tim Burton
 Juan Antonio Ballona
+Alex de la Iglesia
+Pedro Almodovar
```

Práctica 1 – Uso de Git y Git Hub

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (version1)
$ git push origin version1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 341 bytes | 68.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To C:/Users/Manuel/Desktop/S2R2-remoto
 * [new branch]      version1 -> version1
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop
$ git clone S2R2-remoto S2R3
Cloning into 'S2R3'.
done.
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R3 (master)
$ cat directores.txt
Steven Spielberg
Tim Burton
Juan Antonio Ballona

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R3 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R3 (master)
$ git checkout version1
Switched to a new branch 'version1'
Branch 'version1' set up to track remote branch 'version1' from 'origin'.
```

c) Muestra el contenido del fichero directores.txt por la pantalla. Comprueba que se muestran los 5 nombres de directores esperado. Haz una captura de pantalla.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R3 (version1)
$ cat directores.txt
Steven Spielberg
Tim Burton
Juan Antonio Ballona
Alex de la Iglesia
Pedro Almodovar
```

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

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R3 (master)
$ git merge version1
Updating 0b33f7c..08cdc1d
Fast-forward
 directores.txt | 2 ++
1 file changed, 2 insertions(+)
```

Práctica 1 – Uso de Git y Git Hub

e) Sube la rama master actualizada al servidor. Haz una captura de pantalla del resultado del comando.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R3 (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0)
To C:/Users/Manuel/Desktop/S2R2-remoto
0b33f7c..08cdc1d master -> master
```

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

a) Obtén los cambios que hay en el repositorio remoto. Haz una captura de pantalla del resultado del comando utilizado.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (version1)
$ git fetch
From C:/Users/Manuel/Desktop/S2R2-remoto
0b33f7c..08cdc1d master -> origin/master
```

b) Actualiza la rama master local con el contenido de la rama master del repositorio remoto. Haz una captura de pantalla del resultado del comando utilizado.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (master)
$ git pull
Updating 0b33f7c..08cdc1d
Fast-forward
 directores.txt | 2 ++
 1 file changed, 2 insertions(+)
```

c) Comprueba que deben aparecer los 5 nombres de directores esperados.

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/S2R2 (master)
$ cat directores.txt
Steven Spielberg
Tim Burton
Juan Antonio Ballón
Alex de la Iglesia
Pedro Almodóvar
```

3.-GitHub

3.1.-Crea una cuenta en GitHub

3.2.-Añade tu dirección de correo de educación

3.3.-Solicita un descuento para uso educativo en

https://education.github.com/discount_requests/new

3.4.-Haz un fork del repositorio localizado en la siguiente url: <https://github.com/interfacesweb-19-20/semana1-github>



Práctica 1 – Uso de Git y Git Hub

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

README.md	Primer Commit
discos.txt	Cuarto Commit +3 Discos. Close #3
libros.txt	Segundo Commit +3 Libros. Close #1
peliculas.txt	Tercer Commit +3 Películas. Close #2

README.md

Diseño de Interfaces Web

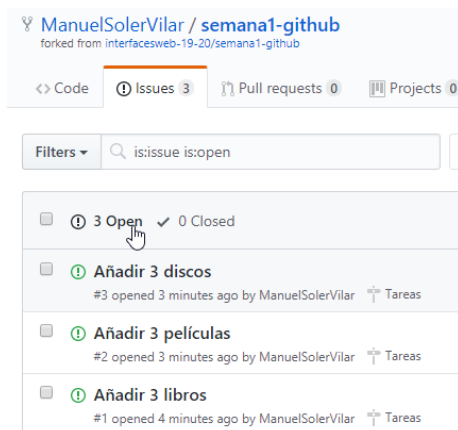
Repositorio para la práctica de uso de GitHub

Incluye a continuación tu nombre y apellidos: Manuel Soler Vilar

b) Crea 3 issues con los siguientes títulos:

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

c) Crea una milestone denominada Tareas 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 de 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.

*Libros.txt

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ nano libros.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git diff
diff --git a/libros.txt b/libros.txt
index d127caf..b22e55e 100644
--- a/libros.txt
+++ b/libros.txt
@@ -1,3 +1,6 @@
-Juego de Tronos
-El Juego de Ender
-La Historia Interminable
+Los Tres Mosqueteros
+De la Tierra a la Luna
+El Conde de Montecristo
```

Práctica 1 – Uso de Git y Git Hub

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   libros.txt

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

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git add .

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   libros.txt
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git commit -m "Segundo Commit +3 Libros. Close #1"
[master ca3a706] Segundo Commit +3 Libros. Close #1
1 file changed, 3 insertions(+)
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 386 bytes | 96.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ManuelSolerVilar/semana1-github.git
4e97d9a..ca3a706 master -> master
```

*Películas.txt

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ nano peliculas.txt
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git diff
diff --git a/peliculas.txt b/peliculas.txt
index 29abf38..cd7f059 100644
--- a/peliculas.txt
+++ b/peliculas.txt
@@ -1,3 +1,6 @@
 Orgullo y Prejuicio
 La boda de mi mejor amiga
 Alta Fidelidad
+Los Goonies
+La Guerra de las Galaxias
+Los Inmortales
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   peliculas.txt

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

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git add .

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   peliculas.txt
```

Práctica 1 – Uso de Git y Git Hub

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git commit -m "Tercer Commit +3 Películas. Close #2"
[master df280bc] Tercer Commit +3 Películas. Close #2
1 file changed, 3 insertions(+)
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 382 bytes | 191.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ManuelSolerVilar/semana1-github.git
ca3a706..df280bc master -> master
```

*Discos.txt

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ nano discos.txt

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git diff
diff --git a/discos.txt b/discos.txt
index 3594f9a..df48d2c 100644
--- a/discos.txt
+++ b/discos.txt
@@ -1,3 +1,6 @@
 The Beatles
 Ella & Louis
 Jagged Little Pill
+Veneno en la Piel
+Descanso Dominical
+Lo Niego Todo
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   discos.txt

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

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git add .

Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Curso Git-Hub/semana1-github (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   discos.txt
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git commit -m "Cuarto Commit +3 Discos. Close #3"
[master fc15805] Cuarto Commit +3 Discos. Close #3
1 file changed, 3 insertions(+)
```

```
Manuel@PC-casa MINGW64 /c/Users/Manuel/Desktop/Git-Hub/semana1-github (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 368 bytes | 92.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ManuelSolerVilar/semana1-github.git
df280bc..fc15805 master -> master
```

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

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

Práctica 1 – Uso de Git y Git Hub