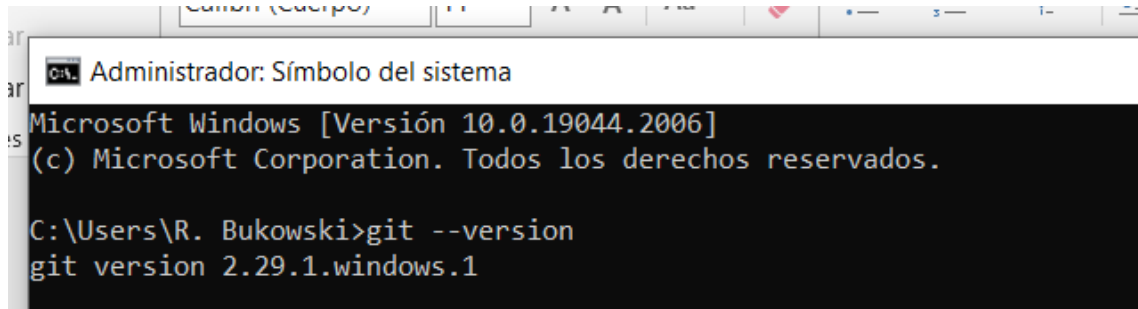


1.- Preparación

Para editar los archivos de texto de las tareas utilizaremos **VISUAL STUDIO CODE**.

2. Fundamentos de Git I

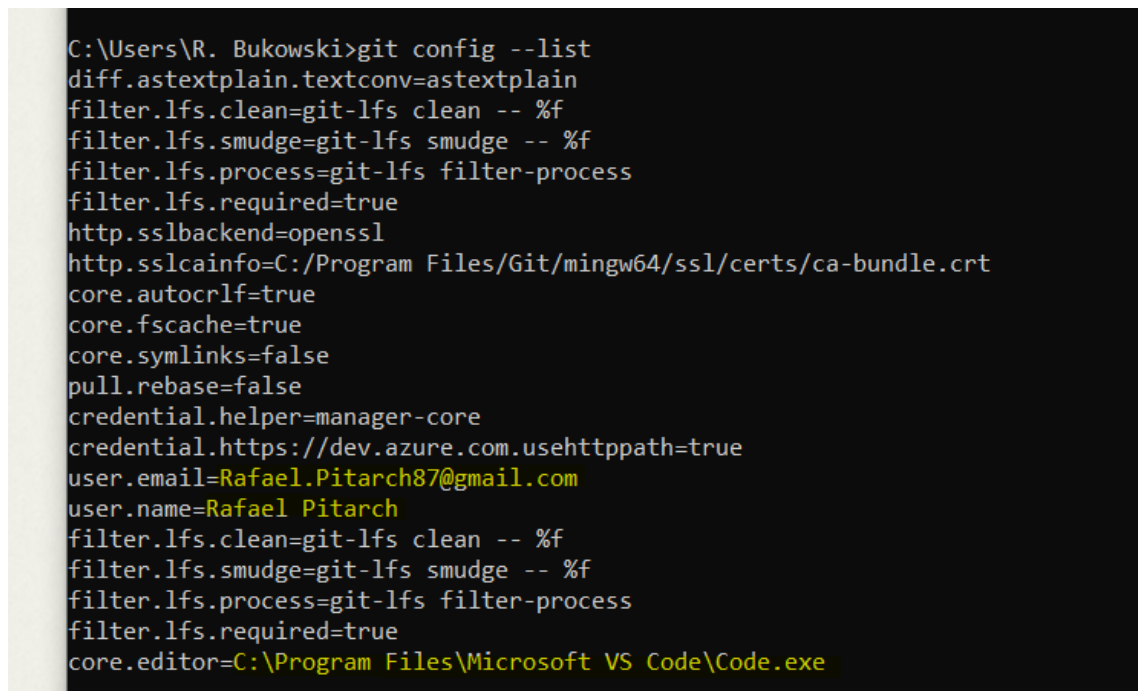
1. Instala Git en tu sistema operativo. Adjunta una captura de pantalla en la que aparezca el resultado de la ejecución del comando `git --version`.



```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2006]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\R. Bukowski>git --version
git version 2.29.1.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.



```
C:\Users\R. Bukowski>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/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
user.email=Rafael.Pitarch87@gmail.com
user.name=Rafael Pitarch
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
core.editor=C:\Program Files\Microsoft VS Code\Code.exe
```

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

a) Crea un repositorio Git.

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

c) Haz un primer commit.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>git commit -m "Primer commit"
[master (root-commit) db715b2] Primer commit
1 file changed, 3 insertions(+)
create mode 100644 libros.txt
```

d) Añade dos libros al archivo libros.txt.

e) Haz un segundo commit.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>git commit -m "Segundo commit"
[master cc4ea80] Segundo commit
1 file changed, 4 insertions(+), 1 deletion(-)
```

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:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    peliculas.txt
```

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

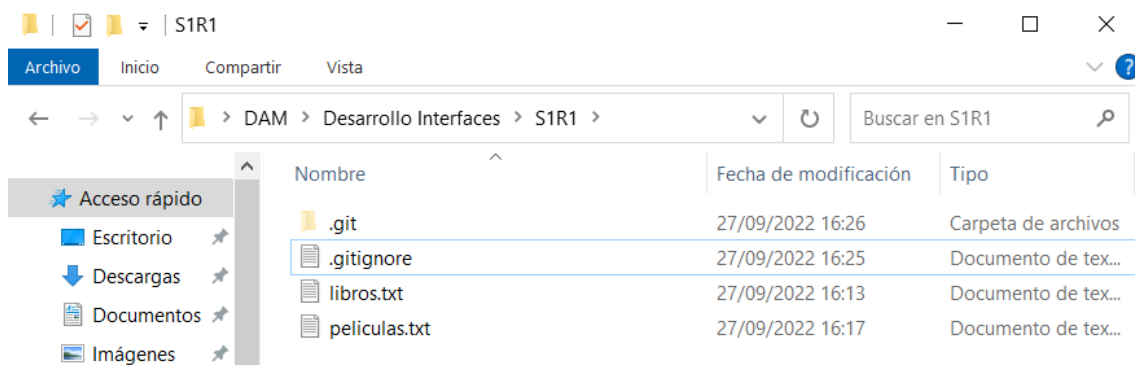
i) Haz un tercer commit que incluya los archivos peliculas.txt y comidas.txt.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>git add comidas.txt peliculas.txt
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>git commit -m "Tercer commit"
[master 0be0b93] Tercer commit
2 files changed, 6 insertions(+)
create mode 100644 comidas.txt
create mode 100644 peliculas.txt
```

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

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

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



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.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>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
```

ñ) Haz un cuarto commit.

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.

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).

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\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:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\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.

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:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\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.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>git add .gitignore  
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>git commit -m "Quinto commit"  
[master 367446e] Quinto commit  
1 file changed, 2 insertions(+)  
create mode 100644 .gitignore
```

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

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S1R1>git log -p  
commit 367446e3f87ea6b734fef04d068800507100aec2 (HEAD -> master)  
Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>  
Date: Tue Sep 27 16:26:37 2022 +0200  
  
    Quinto commit  
  
diff --git a/.gitignore b/.gitignore  
new file mode 100644  
index 0000000..248ed1f  
--- /dev/null  
+++ b/.gitignore  
@@ -0,0 +1,2 @@  
+datos.bak  
+output/salida.txt  
\ No newline at end of file  
  
commit c1e25850db32588817da658ae46ed8d6a777f7da  
Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>  
Date: Tue Sep 27 16:20:42 2022 +0200  
  
    Cuarto commit  
  
diff --git a/comidas.txt b/comidas.txt  
deleted file mode 100644  
index 187d65f..0000000  
--- a/comidas.txt  
+++ /dev/null  
@@ -1,3 +0,0 @@  
-Macarrones con chorizo  
:...skipping...  
commit 367446e3f87ea6b734fef04d068800507100aec2 (HEAD -> master)  
Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>  
Date: Tue Sep 27 16:26:37 2022 +0200  
  
    Quinto commit  
  
diff --git a/.gitignore b/.gitignore  
new file mode 100644  
index 0000000..248ed1f  
--- /dev/null  
+++ b/.gitignore  
@@ -0,0 +1,2 @@  
+datos.bak  
+output/salida.txt  
\ No newline at end of file  
  
commit c1e25850db32588817da658ae46ed8d6a777f7da  
Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>  
Date: Tue Sep 27 16:20:42 2022 +0200
```

3. Fundamentos de Git II

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

- a) Crea un repositorio Git.
- b) Crea un fichero denominado actores.txt. Añade tres nombres de actores cada uno en una línea distinta.
- c) Haz un primer commit.
- d) Crea una rama denominada test.
- e) Cambia a la rama test
- f) En la rama test crea un fichero denominado actrices.txt. Añade tres nombres de actrices y realiza un commit en dicha rama.
- g) Haz una captura de pantalla del resultado del comando git log --graph --all.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R1>git log --graph --all
* commit 7cdd970c043f779cfd8310f4da995aeb54437554 (HEAD -> test)
  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
  Date:   Tue Sep 27 16:40:33 2022 +0200

    Commit en rama

* commit 1b7a524ed3e2420ce378654fa1bf15760877b8ad (master)
  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
  Date:   Tue Sep 27 16:33:17 2022 +0200

    Primer commit - archivo actores

C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R1>
```

- h) Cambia a la rama 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.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R1>git merge test
Updating 1b7a524..7cdd970
Fast-forward
 actores.txt | 3 +++
 1 file changed, 3 insertions(+)
 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.
- k) En la rama master, añade una actriz al fichero actrices.txt y haz un commit.
- l) Cambia a la rama test2
- m) En la rama test2, añade una actriz al fichero actrices.txt y haz otro commit.

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`.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R1>git log --graph --all
* commit c1d96cad1166162b8d301c6f4ad8b82e84d0d0b5 (HEAD -> test2)
  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
  Date: Tue Sep 27 16:54:47 2022 +0200

    Tercer commit, a\u00f1ado otra actriz

* commit f37a0310d02079c4e43a8c16fc7bb2f51247702d (master)
  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
  Date: Tue Sep 27 16:53:05 2022 +0200

    Segundo commit, a\u00f1ado actriz

* commit 7cdd970c043f779cfd8310f4da995aeb54437554 (test)
  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
  Date: Tue Sep 27 16:40:33 2022 +0200

    Commit en rama

* commit 1b7a524ed3e2420ce378654fa1bf15760877b8ad
  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
  Date: Tue Sep 27 16:33:17 2022 +0200

    Primer commit - archivo actores
```

ñ) Cambia a la rama `master`

o) Incorpora los cambios de la rama `test2` a la rama `master`. \u00bfSe produce un conflicto? De ser as\u00ed realiza una captura del comando `git status`.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R1>git merge test2 -m "fusiono cambios"
Auto-merging actrices.txt
CONFLICT (content): Merge conflict in actrices.txt
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\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

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

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.

```

C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R1>git log --graph --all
*   commit 3a88349a59ab1a6b07b6e1767d3367c8d7873b62 (HEAD -> master)
|  Merge: f37a031 c1d96ca
|  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
|  Date:   Tue Sep 27 17:01:07 2022 +0200
|
|      Cuarto commit - arreglado el conflicto
|
| *   commit c1d96cad1166162b8d301c6f4ad8b82e84d0d0b5 (test2)
| |  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
| |  Date:   Tue Sep 27 16:54:47 2022 +0200
| |
| |      Tercer commit, a ado otra actriz
| |
| *   commit f37a0310d02079c4e43a8c16fc7bb2f51247702d
| |  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
| |  Date:   Tue Sep 27 16:53:05 2022 +0200
| |
| |      Segundo commit, a ado actriz
| |
| *   commit 7cdd970c043f779cfd8310f4da995aeb54437554 (test)
| |  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
| |  Date:   Tue Sep 27 16:40:33 2022 +0200
| |
| |      Commit en rama
| |
| *   commit 1b7a524ed3e2420ce378654fa1bf15760877b8ad
| |  Author: Rafael Pitarch <Rafael.Pitarch87@gmail.com>
| |  Date:   Tue Sep 27 16:33:17 2022 +0200
| |
| |      Primer commit - archivo actores

```

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

```

C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces>git clone "S2R2 - remoto" S2R2
Cloning into 'S2R2'...
done.

```

A continuaci n, realiza las siguientes acciones en el repositorio S2R2:

- a) Crea un archivo denominado `directores.txt`. A ade el nombre de tres directores de cine.
- b) Haz un commit.
- c) Realiza un push al repositorio remoto. Adjunta captura de pantalla del resultado.

```

C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R2>git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 266 bytes | 133.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R2 - remoto
 * [new branch]      master -> master

```

d) Crea una rama denominada version1.

e) Cambia a la rama version1.

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.

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.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R2>git push --set-upstream origin version1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R2 - remoto
 * [new branch]      version1 -> version1
Branch 'version1' set up to track remote branch 'version1' from 'origin'.
```

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.

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

Comprueba que se crea una rama local version1 con el contenido de la rama remota origin/version1 y enlazada con ella. Al

clonar el repositorio la rama no existía (solo se clona la rama principal, master), pero al cambiar a una rama que existe en el remoto se produce su creación local y enlazado con su correspondiente remota.

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.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R3>type directores.txt
Guy Ritchie
Woody Allen
Pedro Almodovar
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R3>git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R3>git checkout version1
Switched to a new branch 'version1'
Branch 'version1' set up to track remote branch 'version1' from 'origin'.

C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R3>type directores.txt
Guy Ritchie
Woody Allen
Pedro Almodovar

Peter Jackson
Steven Spielberg
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R3>
```


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.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R3>git merge version1 -m "fusiono cambios en directores"
Updating a777ede..af504df
Fast-forward (no commit created; -m option ignored)
 directores.txt | 5 +++-
 1 file changed, 4 insertions(+), 1 deletion(-)

C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R3>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R2 - remoto
 a777ede..af504df 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 sin fusionarlos en la rama local. Adjunta captura de pantalla del resultado del comando utilizado.

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R2>git fetch
From C:/Users/R. Bukowski/Desktop/DAM/Desarrollo Interfaces/S2R2 - remoto
 1b05b02..b5be76b master -> origin/master
```

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:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R2>git pull
```

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

```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\S2R2>type directores.txt
Pedro Almodovar
Steven Spielberg
Woody Allen
Peter Jackson
Martin Scorsese
```

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

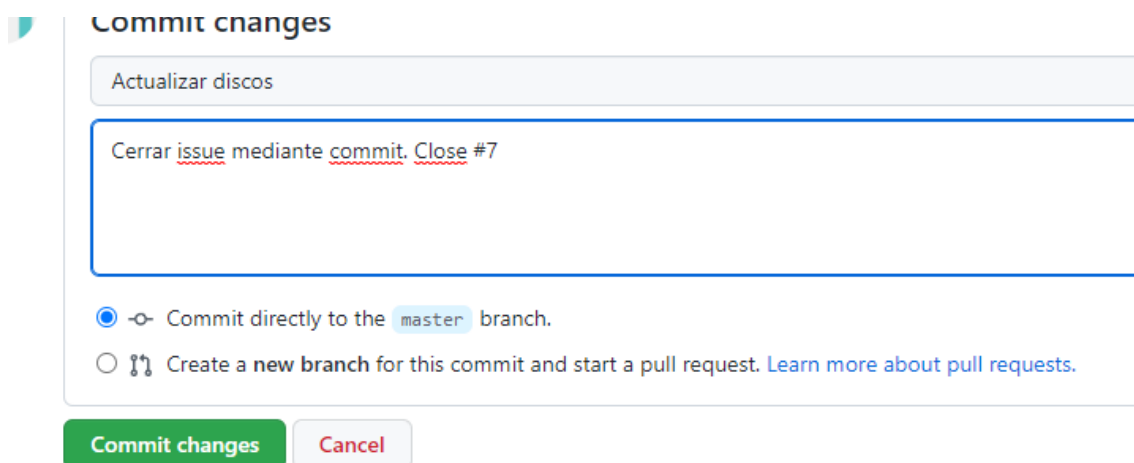
b) Crea 3 issues con los siguientes títulos. Si no ves la pestaña de issues, actívala desde los ajustes (settings) del repositorio.

- 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.



```
C:\Users\R. Bukowski\Desktop\DAM\Desarrollo Interfaces\DesarrolloInterfaces>git commit -m "Cerrar issue. Close #4"
```

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.

