**Ejercicio 1**

**Configurar Git definiendo el nombre del usuario, el correo electrónico y activar el coloreado de la salida. Mostrar la configuración final.**

git config - -global user.name “serporion”

git config - -global user.email “serporionGit@hotmail.com”

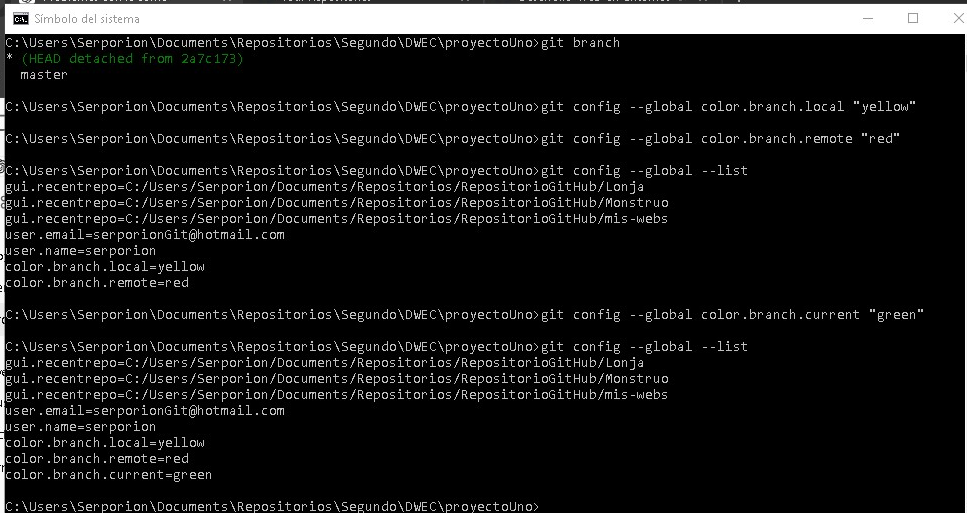
git config --global --list

git config --global color.branch.current "green"

git config --global color.branch.local "yellow"

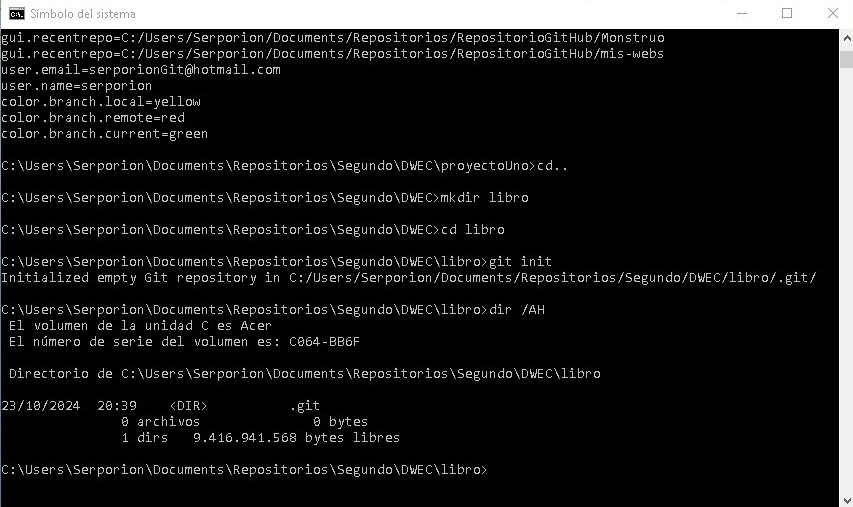
git config --global color.branch.remote "red"

git config - -global - -list



## Ejercicio 2

**Crear un repositorio nuevo con el nombre libro y mostrar su contenido.**



## Ejercicio 3

1. **Comprobar el estado del repositorio.**

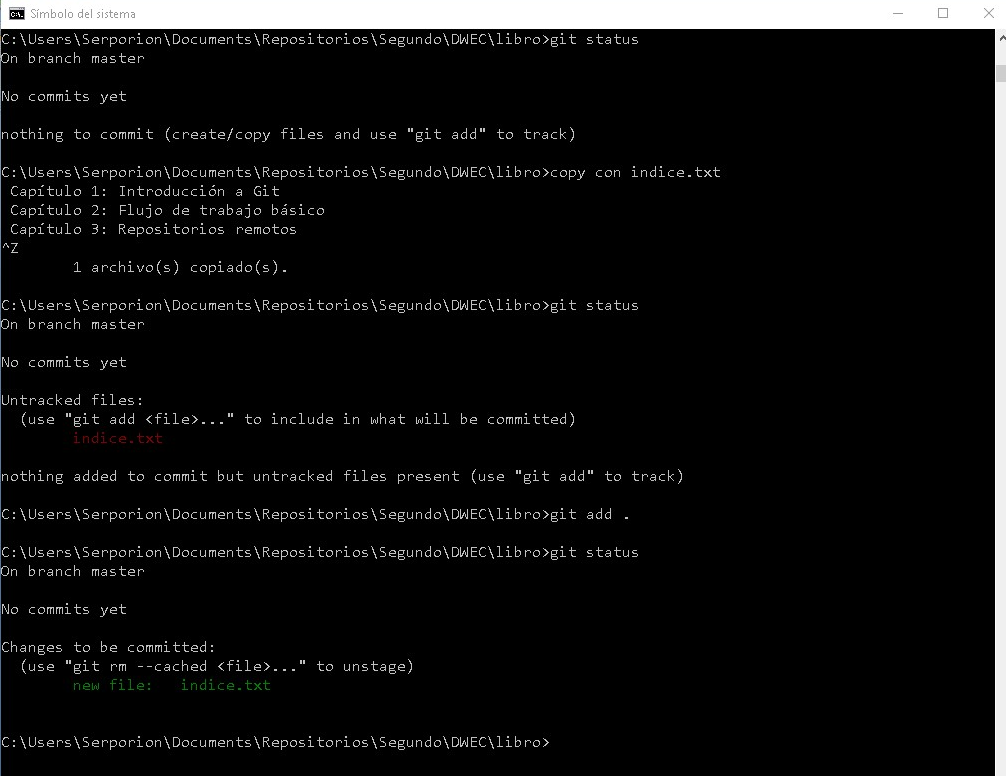
git status

1. **Crear un fichero indice.txt con el siguiente contenido:**

copy con indice.txt

1. **Comprobar de nuevo el estado del repositorio.**
2. **Añadir el fichero a la zona de intercambio temporal.**

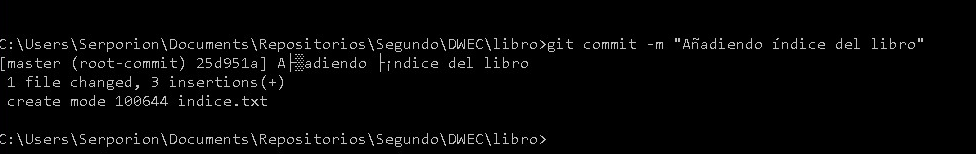
git add .

1. **Volver a comprobar una vez más el estado del repositorio.**

## Ejercicio 4

**Realizar un commit de los últimos cambios con el mensaje “Añadido índice del libro.” y ver el estado del repositorio.**

git commit -m "Añadiendo índice del libro"



**Ejercicio 5**

1. **Cambiar el fichero indice.txt para que contenga el capítulo 3, desplazando el antiguo capítulo 3 al capítulo 4.**
2. **Mostrar los cambios con respecto a la última versión guardada en el repositorio**

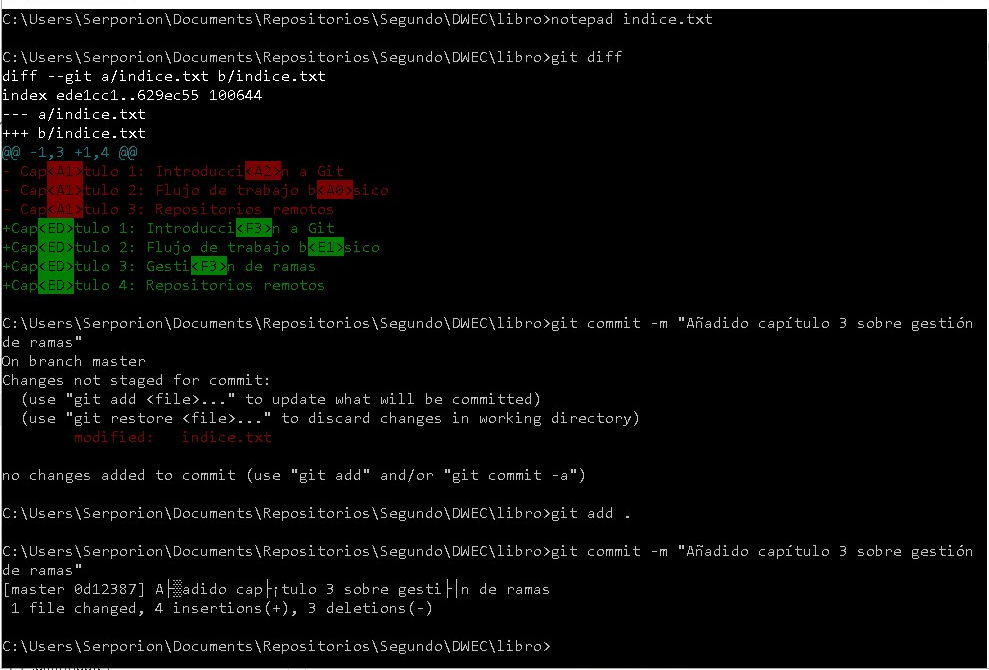
git diff (si queremos ver la diferencias cuando no hemos hecho un add)

git diff - -cached (si queremos ver la diferencias cuando ya hemos hecho un add)

git diff HEAD (si queremos ver cambios tanto con add como sin add)

1. **Hacer un commit de los cambios con el mensaje “Añadido capítulo 3 sobre gestión de ramas”.**

git add .

git commit –m "“Añadido capítulo 3 sobre gestión de ramas”

**Ejercicio 6**

1. **Mostrar los cambios de la última versión del repositorio con respecto a la anterior.**

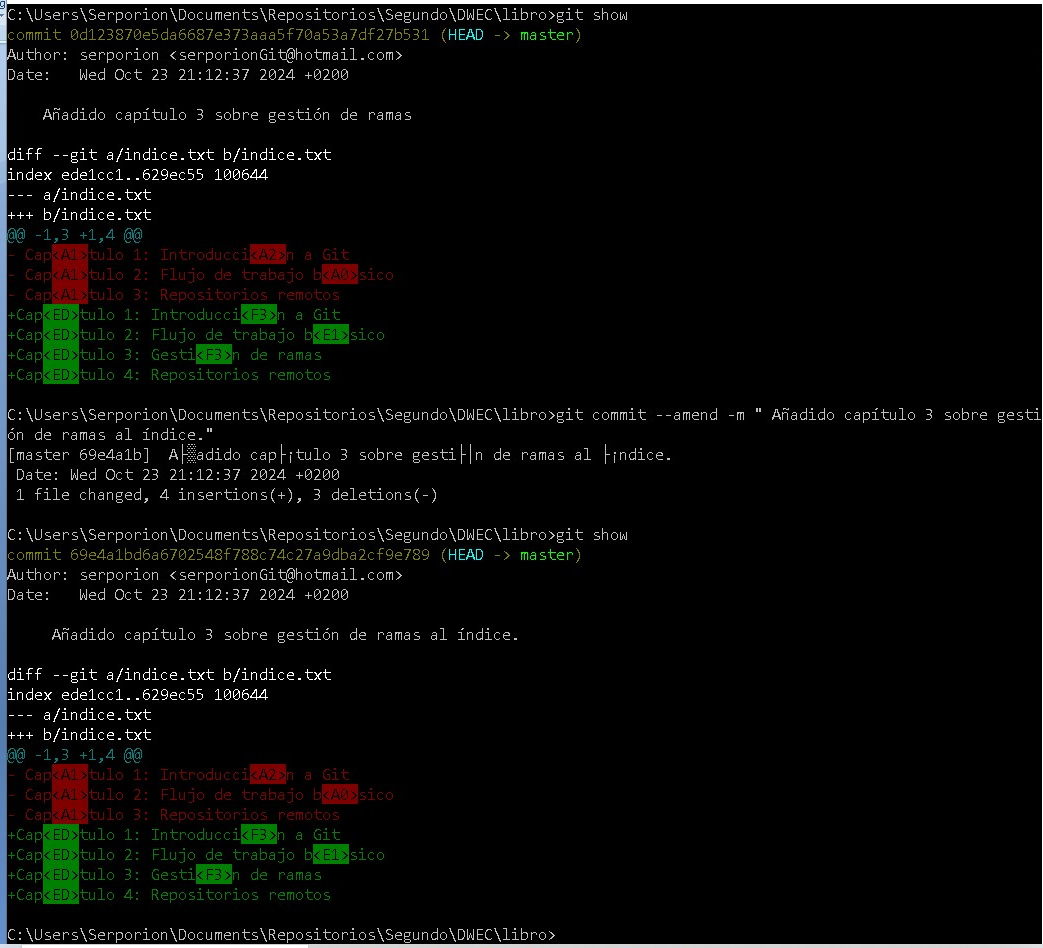
git show

1. **Cambiar el mensaje del último commit por “Añadido capítulo 3 sobre gestión de ramas al índice.”**

git commit --amend -m " Añadido capítulo 3 sobre gestión de ramas al índice."

1. **Volver a mostrar los últimos cambios del repositorio.**

git show



**Ejercicio 7**

1. **Mostrar el historial de cambios del repositorio.**

git log

1. **Crear la carpeta capitulos y crear dentro de ella el fichero capitulo1.txt con el siguiente texto.**

echo Git es un sistema de control de versiones ideado por Linus Torvalds. > capitulos\capitulo1.txt

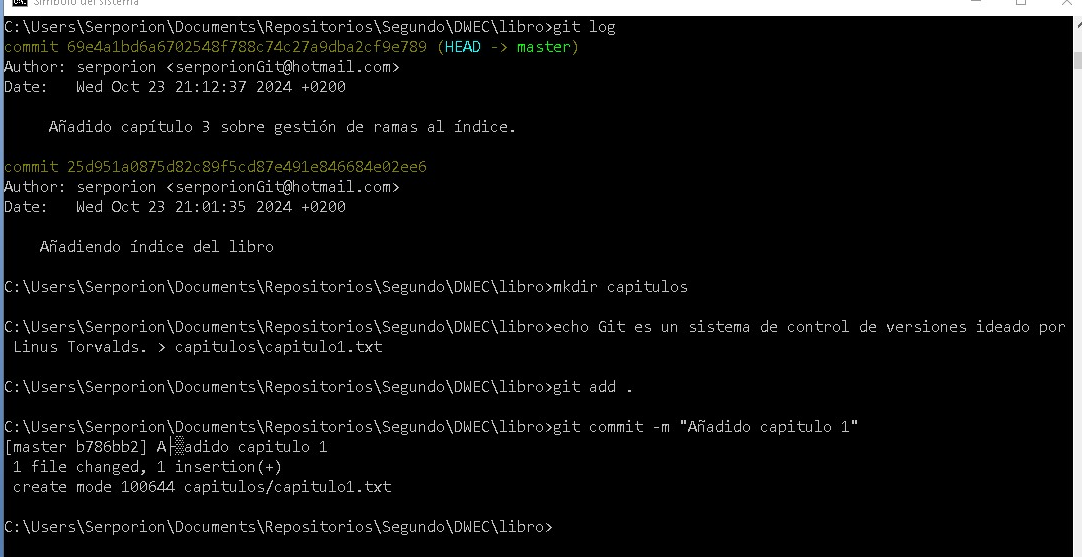
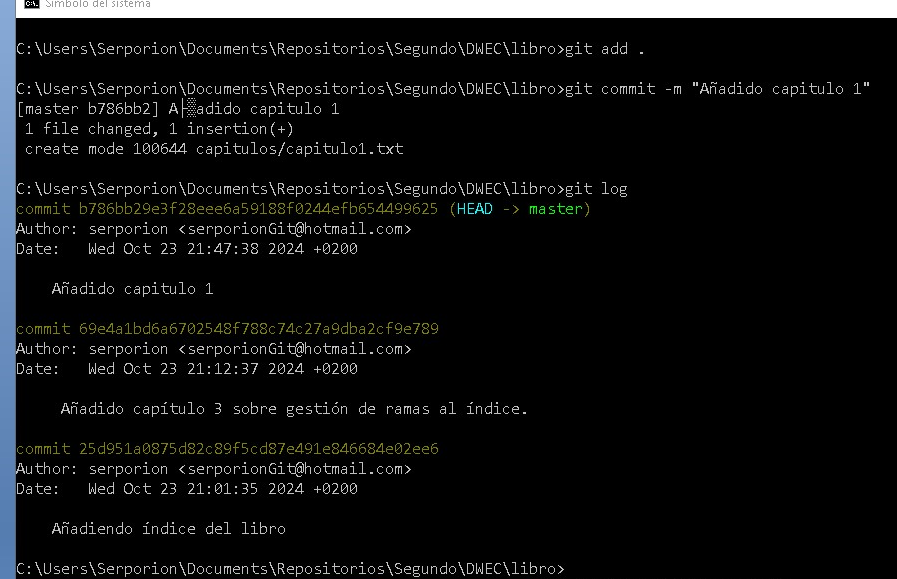
1. **Añadir los cambios a la zona de intercambio temporal.**

git add .

1. **Hacer un commit de los cambios con el mensaje “Añadido capítulo 1.”**

git commit –m “Añadido capítulo 1”

1. **Volver a mostrar el historial de cambios del repositorio.**



**Ejercicio 8**

1. **Crear el fichero capitulo2.txt en la carpeta capitulos con el siguiente texto.**

notepad capitulos\capitulo2.txt

1. **Añadir los cambios a la zona de intercambio temporal.**

git add .

1. **Hacer un commit de los cambios con el mensaje “Añadido capítulo 2.”**

git commit –m “Añadido capitulo2”

1. **Mostrar las diferencias entre la última versión y dos versiones anteriores.**

git diff HEAD~2 HEAD



**Ejercicio 9**

1. **Crear el fichero capitulo3.txt en la carpeta capitulos con el siguiente texto.**

echo Git permite la creación de ramas lo que permite tener distintas versiones del mismo proyecto y trabajar de manera simultanea en ellas. > capitulos\capitulo3.txt

1. **Añadir los cambios a la zona de intercambio temporal.**

git add .

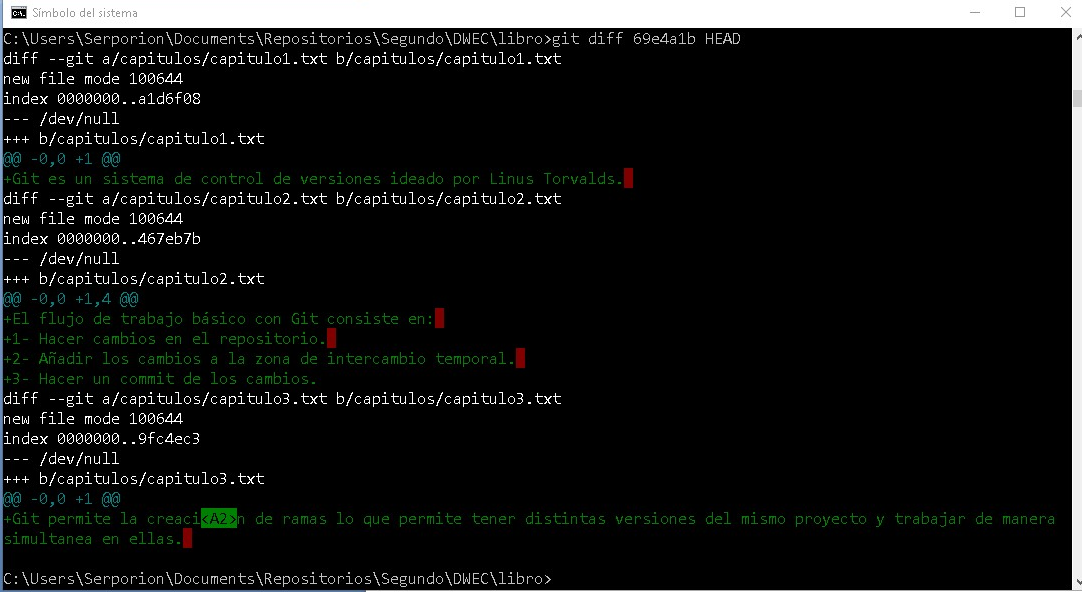
1. **Hacer un commit de los cambios con el mensaje “Añadido capítulo 3.”**

git commit –m “Añadido capítulo 3”

1. **Mostrar las diferencias entre la primera y la última versión del repositorio.**

git log - -oneline (copiamos el hash abreviado del commit)

git diff “hash del primero” HEAD (Aquí está apuntando al último commit).

git diff 69e4a1b HEAD

**Ejercicio 10**

1. **Añadir al final del fichero indice.txt la siguiente línea:**

notepad indice.txt

1. **Añadir los cambios a la zona de intercambio temporal.**

git add .

1. **Hacer un commit de los cambios con el mensaje “Añadido capítulo 5 al índice.”.**

git commit –m “Añadido capítulo 5 al índice”

1. **Mostrar quién ha hecho cambios sobre el fichero indice.txt.**

git log --follow -- indice.txt (dejar espacio entre - - e índice)

**Ejercicio 11**

1. **Eliminar la última línea del fichero indice.txt y guardarlo.**

notepad índice.txt

1. **Comprobar el estado del repositorio.**

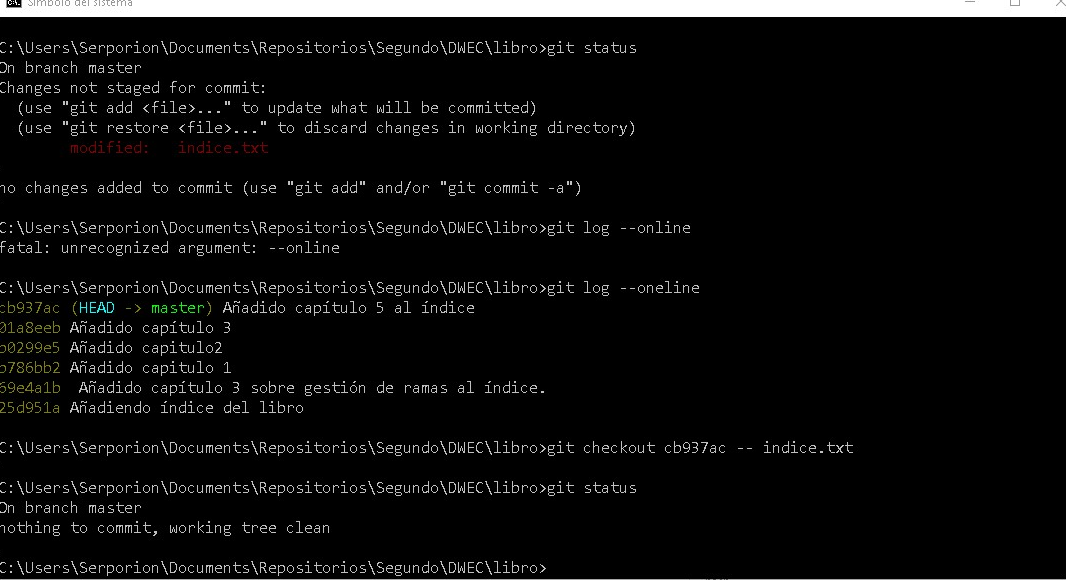
git status

1. **Deshacer los cambios realizados en el fichero indice.txt para volver a la versión anterior del fichero.**

git log –online (elegimos el hash abreviado del commit)

git checkout “hash abreviado” - - índice.txt

1. **Volver a comprobar el estado del repositorio.**

git status

**Ejercicio 12**

1. **Eliminar la última línea del fichero indice.txt y guardarlo.**

notepad índice.txt

1. **Añadir los cambios a la zona de intercambio temporal.**

git add .

1. **Comprobar de nuevo el estado del repositorio.**

git status

1. **Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.**

git reset índice.txt (si se indica un fichero, solo ese será afectado)

1. **Comprobar de nuevo el estado del repositorio.**

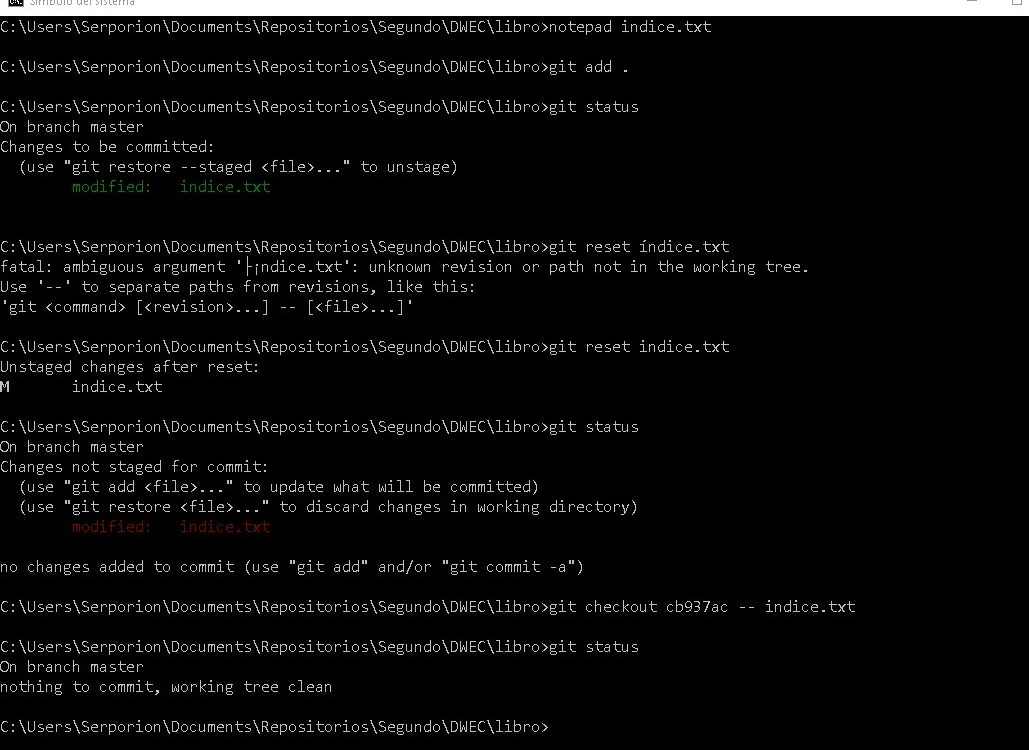
git status

1. **Deshacer los cambios realizados en el fichero indice.txt para volver a la versión anterior del fichero.**

git log –online (elegimos el hash abreviado del commit)

git checkout “hash abreviado” - - índice.txt

1. **Volver a comprobar el estado del repositorio.**

git status

**Ejercicio 13**

1. **Eliminar la última línea del fichero indice.txt y guardarlo.**

notepad índice.txt

1. **Eliminar el fichero capitulos/capitulo3.txt.**

del capitulos\capitulo3.txt

1. **Añadir un fichero nuevo captitulos/capitulo4.txt vacío.**

copy con capitulos\capitulo4.txt

1. **Añadir los cambios a la zona de intercambio temporal.**

git add .

1. **Comprobar de nuevo el estado del repositorio.**

git status

1. **Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.**

git reset

1. **Comprobar de nuevo el estado del repositorio.**

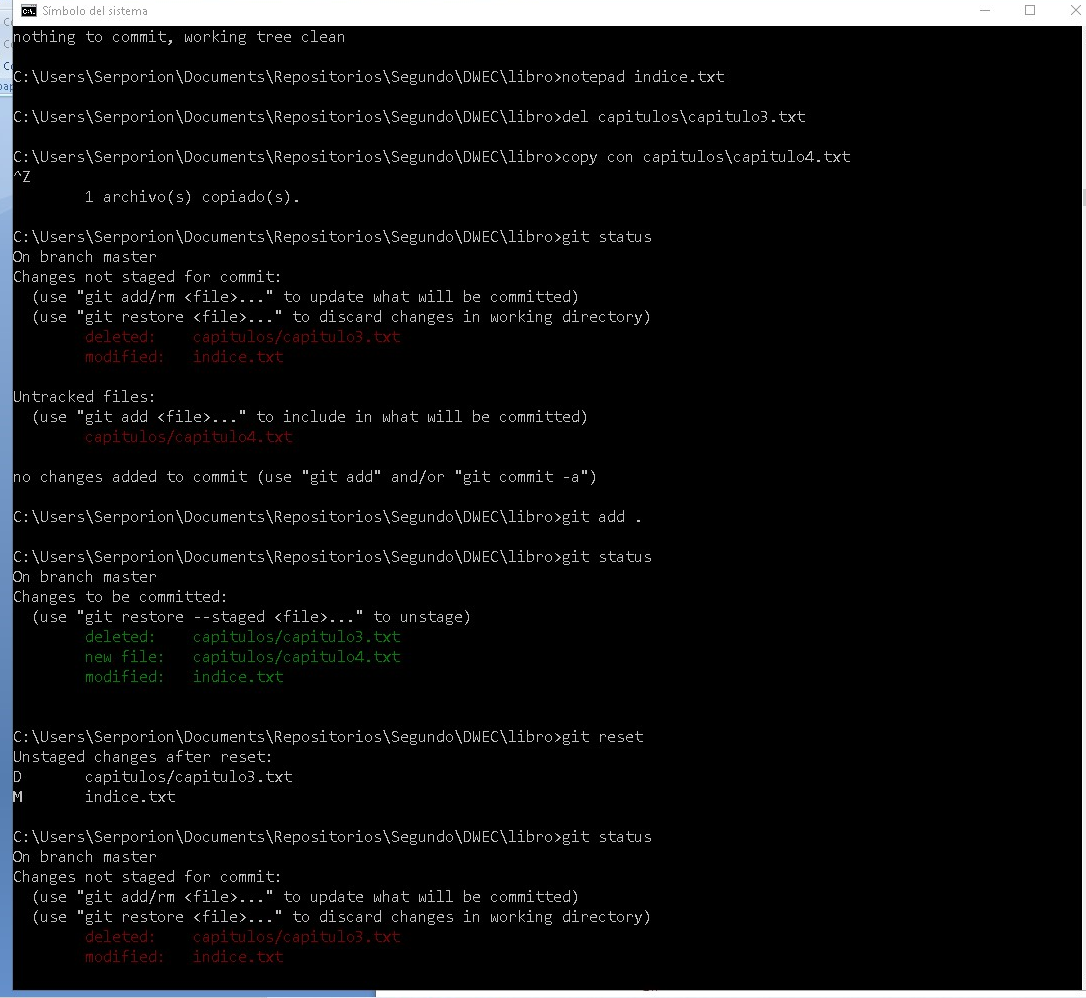
git status

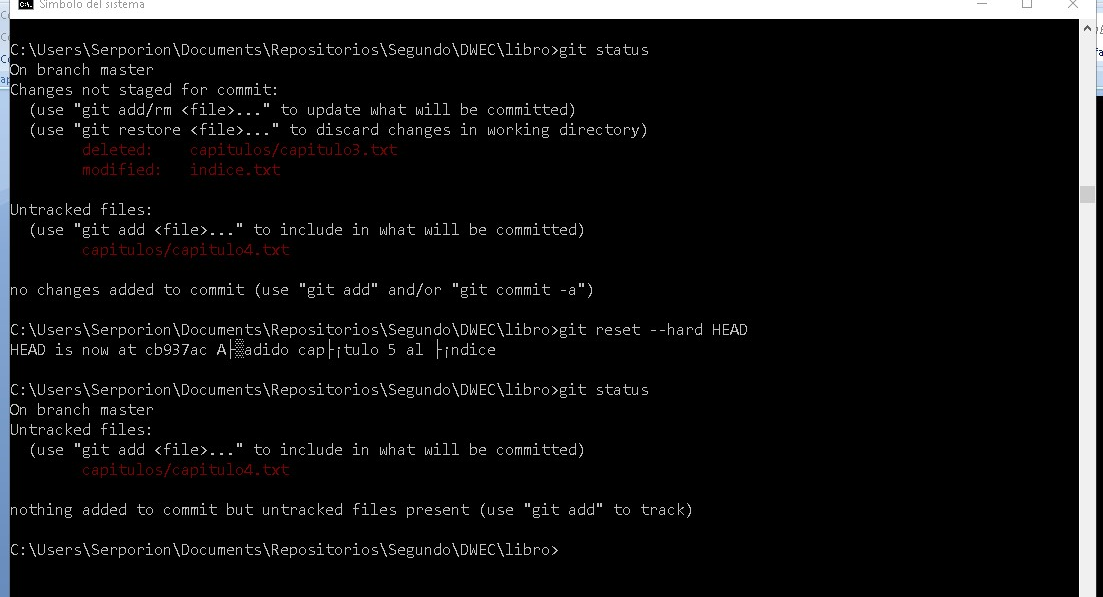
1. **Deshacer los cambios realizados para volver a la versión del repositorio.**

git reset --hard “hash” o git reset --hard HEAD ( si coinciden)

1. **Volver a comprobar el estado del repositorio.**

git status





Git al no tener seguimiento de ese fichero capitulo4.txt, no sabe qué hacer con él. El no lo borra. Lo borramos nosotros a mano para seguir con los ejercicios.

**Ejercicio 14**

1. **Eliminar la última línea del fichero indice.txt y guardarlo.**

Notepad índice.txt

1. **Eliminar el fichero capitulos/capitulo3.txt.**

del capítulos\capitulo3.txt

1. **Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Borrado accidental.”**

git add . y git commit –m “Borrado accidental”

1. **Comprobar el historial del repositorio.**

git log --oneline

1. **Deshacer el último commit pero mantener los cambios anteriores en el directorio de trabajo y la zona de intercambio temporal.**

git reset --soft HEAD~1

1. **Comprobar el historial y el estado del repositorio.**

git log –oneline y git status

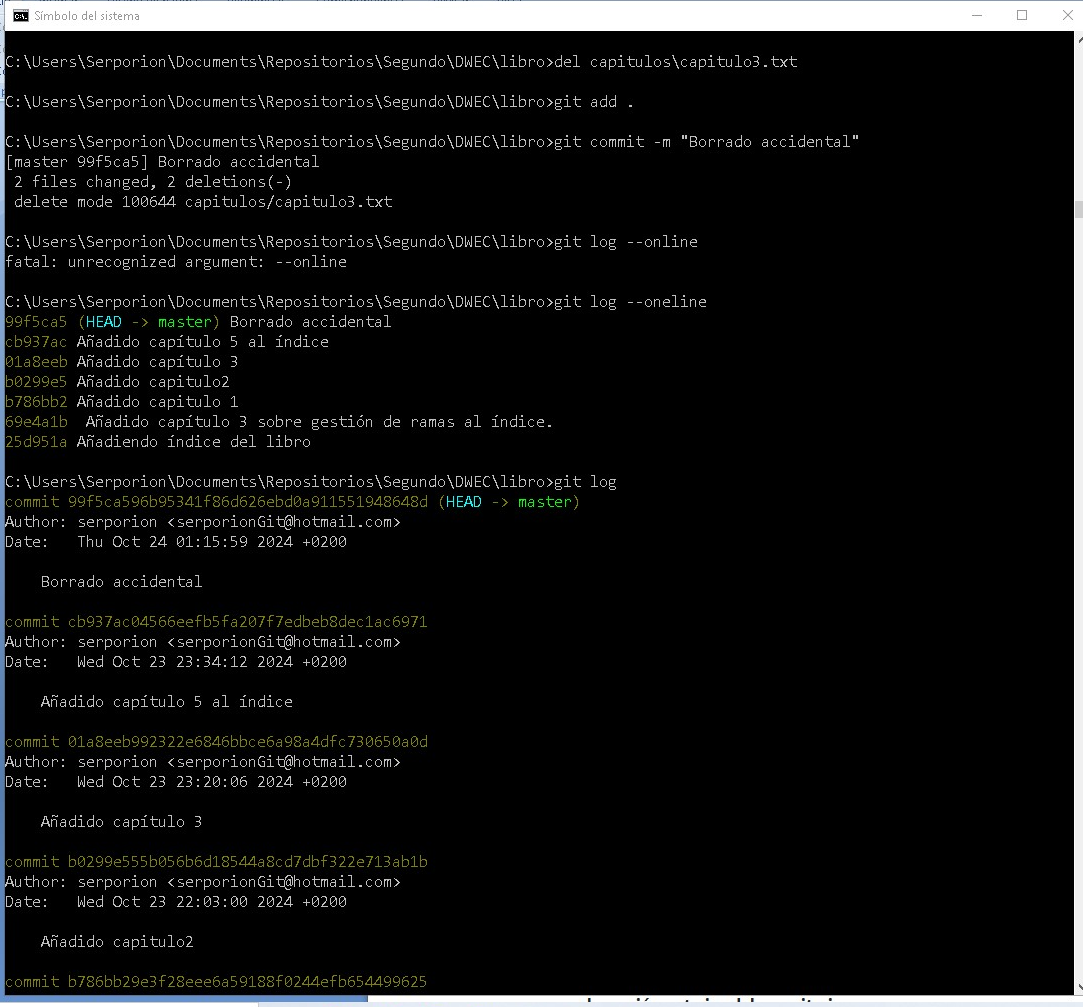
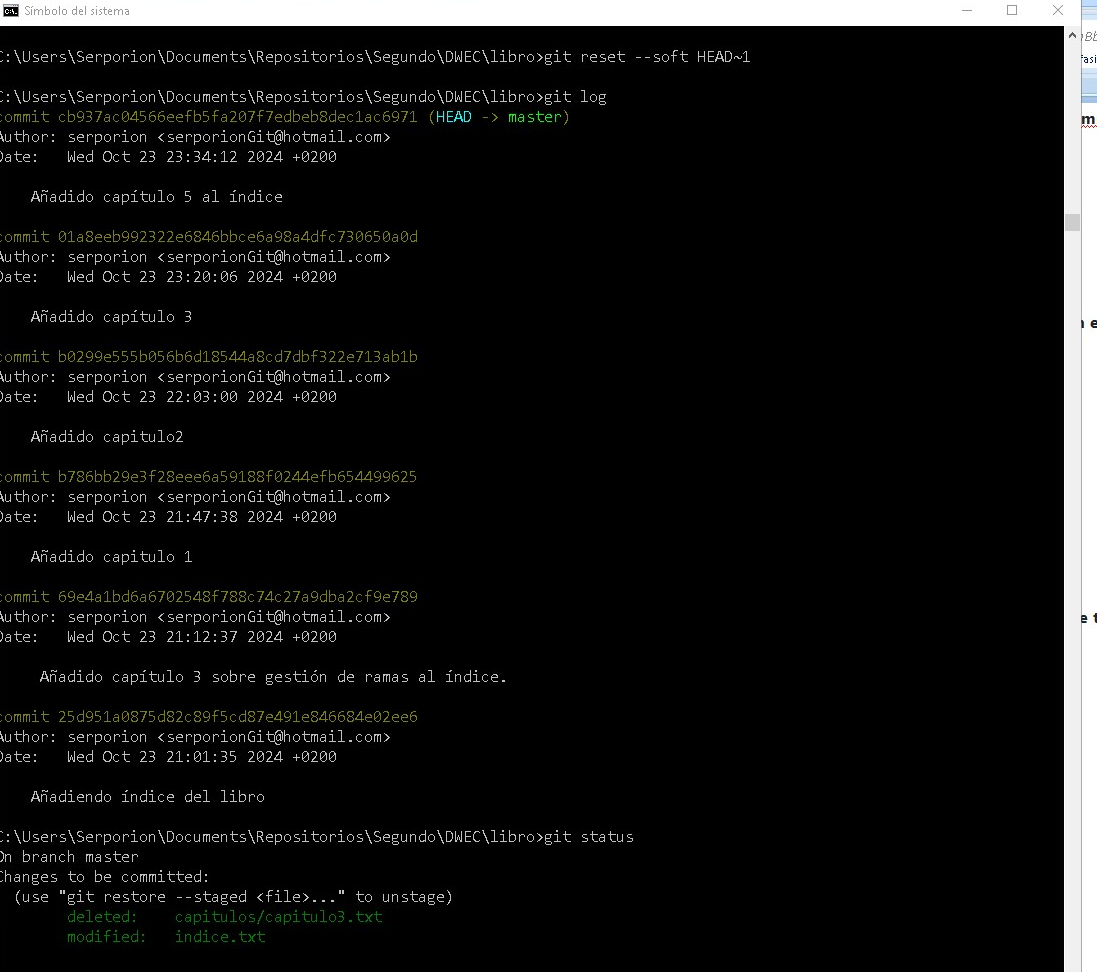
1. **Volver a hacer el commit con el mismo mensaje de antes.**

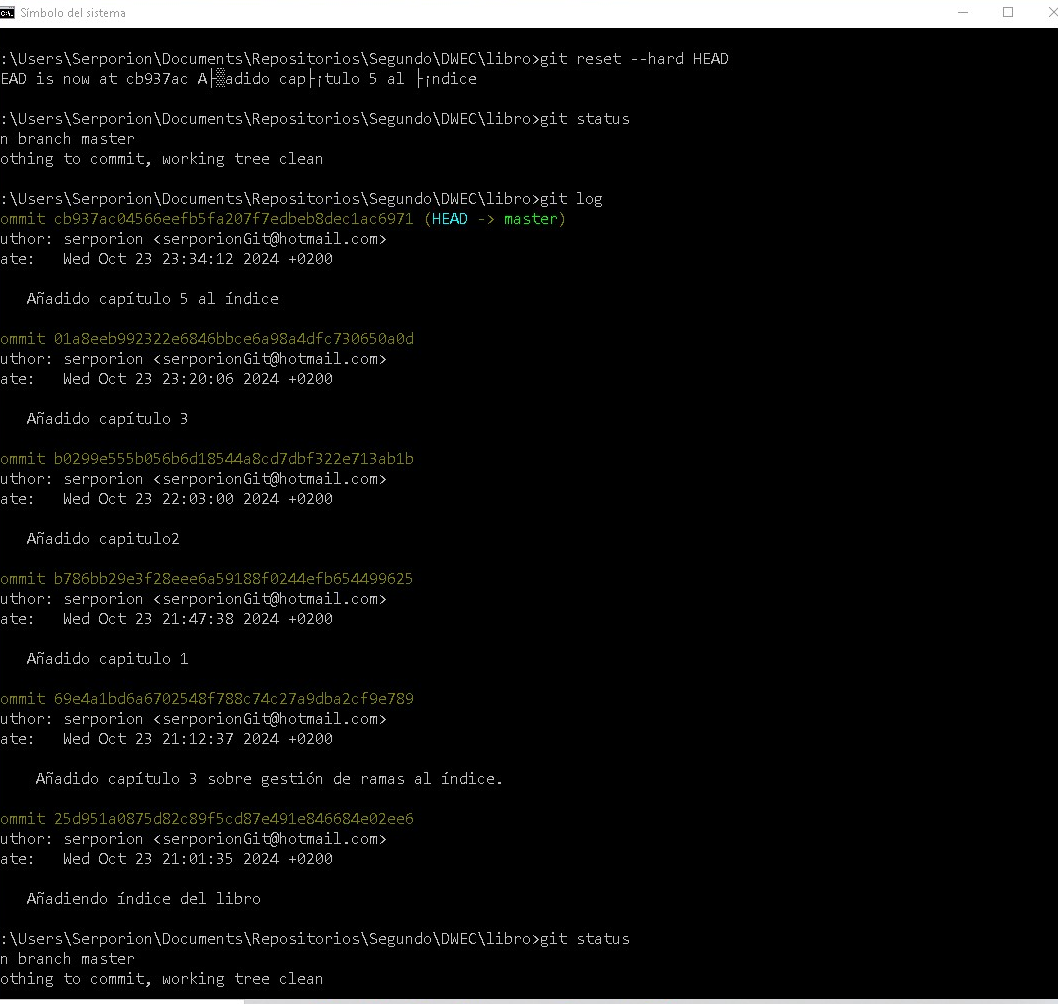
git commit –m “Borrado accidental”

1. **Deshacer el último commit y los cambios anteriores del directorio de trabajo volviendo a la versión anterior del repositorio.**

git reset - -hard HEAD

1. **Comprobar de nuevo el historial y el estado del repositorio.**

****



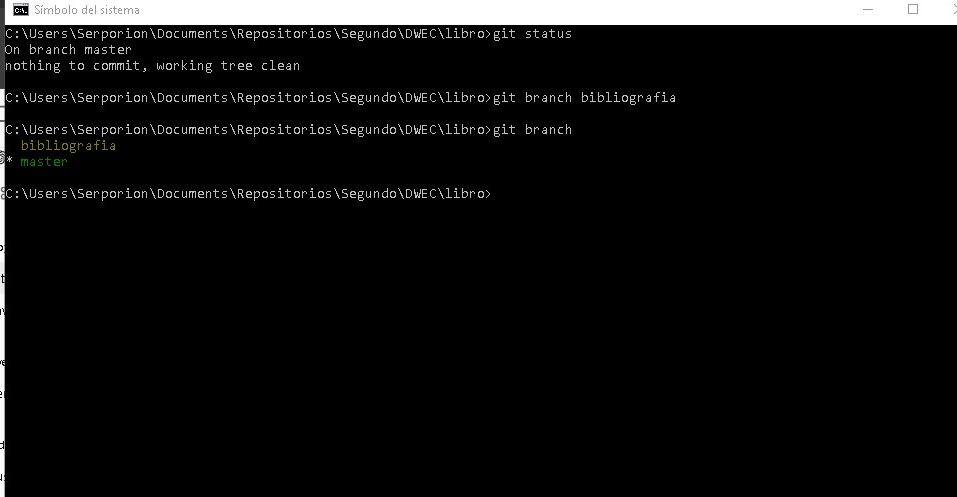
**Ejercicio 15**

**Crear una nueva rama bibliografía y mostrar las ramas del repositorio.**

git branch bibliografía

git branch (ordena las ramas por orden alfabético en la salida).

(git checkout -b bibliografia: Crea también la rama yte cambia a ella. El capítulo4.txt se hubiera referenciado allí. git branch bibliografía crea la rama pero se queda apuntando al último commit).



**Ejercicio 16**

1. **Crear el fichero capitulos/capitulo4.txt y añadir el texto siguiente**

echo “En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto” > capitulos\capitulo4.txt

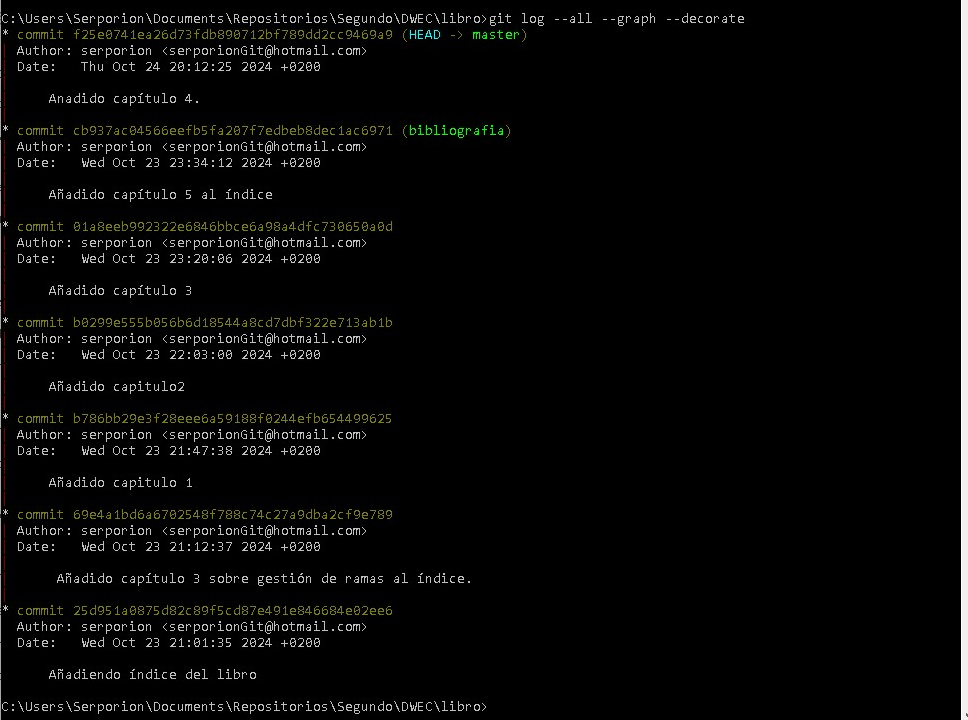
1. **Añadir los cambios a la zona de intercambio temporal.**

git add .

1. **Hacer un commit con el mensaje “Añadido capítulo 4.”**

git commit –m “Anadido capítulo 4.”

1. **Mostrar la historia del repositorio incluyendo todas las ramas.**

git log --oneline --all --graph –decorate

**Ejercicio 17**

1. **Cambiar a la rama bibliografia.**

git checkout bibliografia

1. **Crear el fichero bibliografia.txt y añadir la siguiente referencia**

echo “Chacon, S. and Straub, B. Pro Git. Apress.” > bibliografia.txt

1. **Añadir los cambios a la zona de intercambio temporal.**

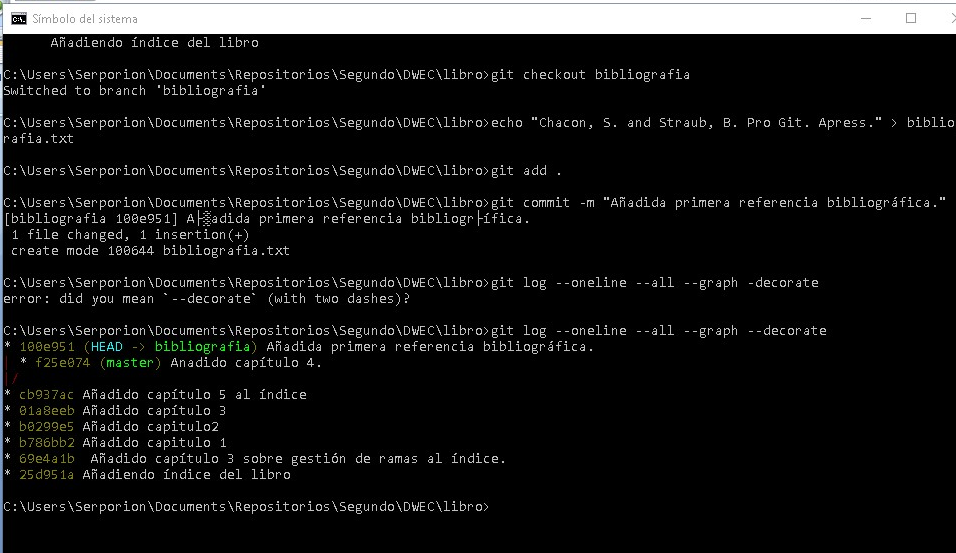
git add .

1. **Hacer un commit con el mensaje “Añadida primera referencia bibliográfica.”**

git commit –m “Añadida primera referencia bibliográfica.”

1. **Mostrar la historia del repositorio incluyendo todas las ramas.**

git log --oneline --all --graph –decorate

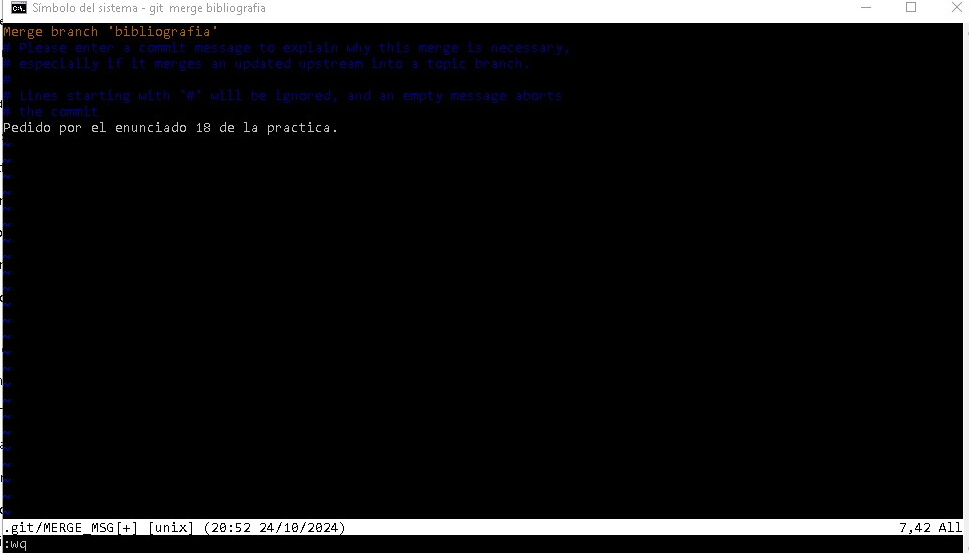
****

**Ejercicio 18**

1. **Fusionar la rama bibliografia con la rama master.**

git checkout master (nos colocamos en la rama que mande)

git merge bibliografía (le decimos que rama nos traemos a la rama que manda)

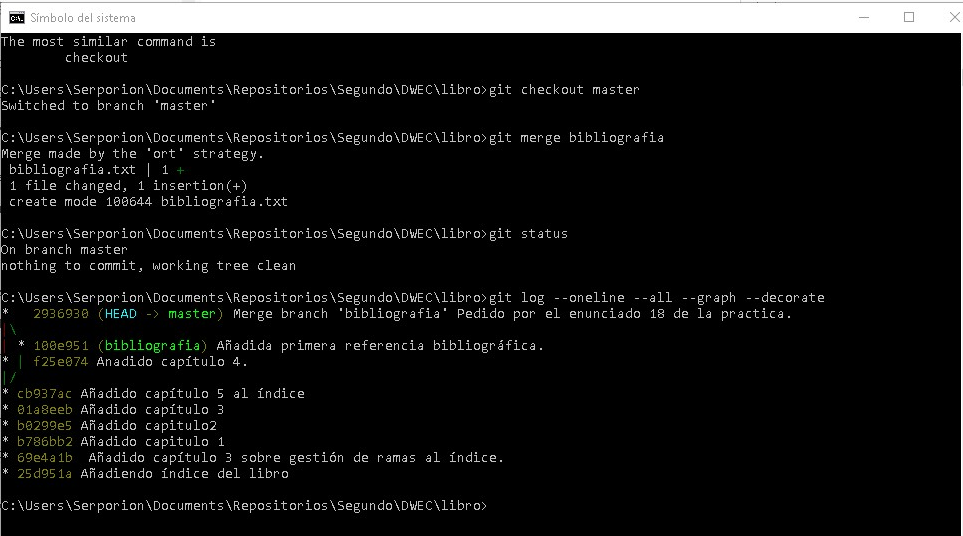


Se abre Vim, y procedemos a escribir lo que nos pide, que es un comentario del porqué hacemos esto. Si hubiéramos escrito lo siguiente no entraríamos a dicho editor:

git merge bibliografia -m "Mensaje de fusión: fusionando bibliografia en master"

1. **Mostrar la historia del repositorio incluyendo todas las ramas.**

git log --oneline --all --graph –decorate

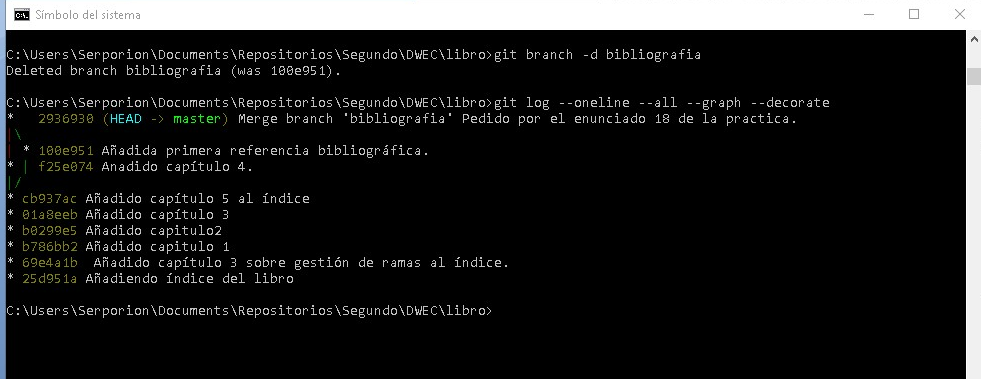
****

1. **Eliminar la rama bibliografia.**

git branch -d bibliografía

1. **Mostrar de nuevo la historia del repositorio incluyendo todas las ramas.**

git log --oneline --all --graph –decorate

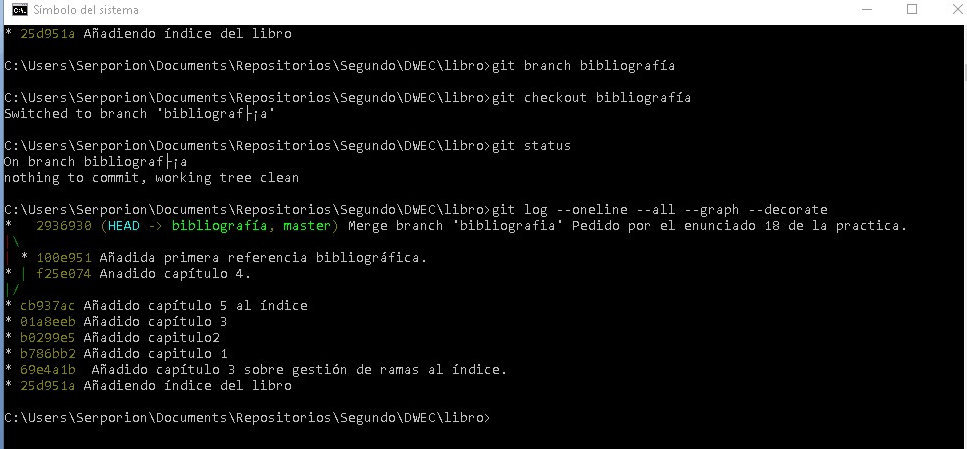
****

**Ejercicio 19**

1. **Crear la rama bibliografia.**

git branch bibliografía

1. **Cambiar a la rama bibliografia.**

git checkout bibligrafia

1. **Cambiar el fichero bibliografia.txt para que contenga las siguientes referencias:**

notepad bibliografia.txt

1. **Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Añadida nueva referencia bibliográfica.”**

git add .

git commit –m “Añadida nueva referencia bibliográfica.”

1. **Cambiar a la rama master.**

git checkout master

1. **Cambiar el fichero bibliografia.txt para que contenga las siguientes referencias:**

notepad bibliografia.txt

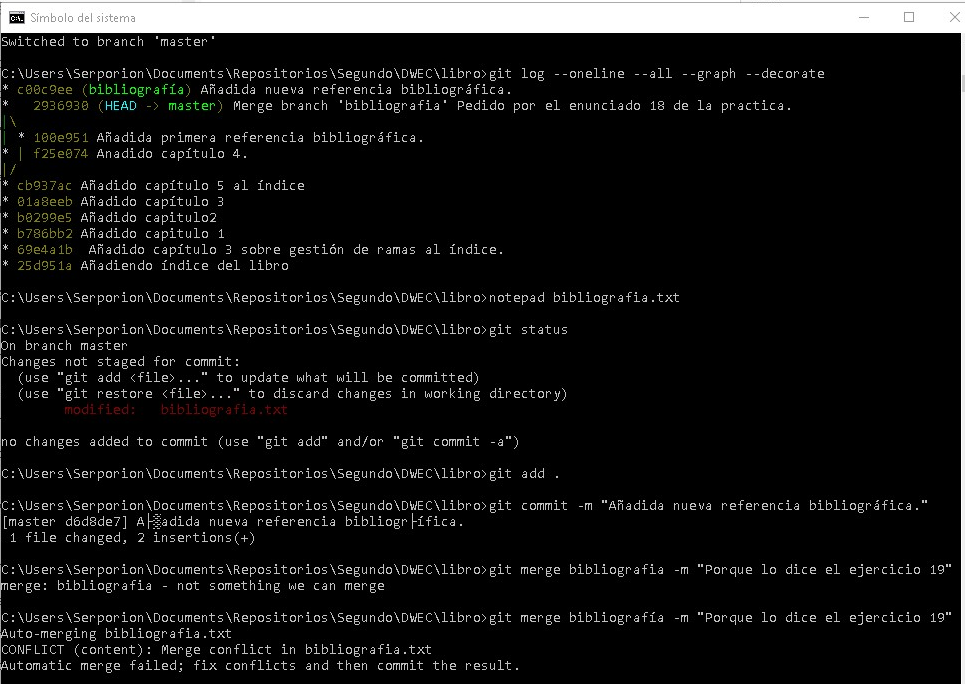
1. **Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Añadida nueva referencia bibliográfica.”**

git add .

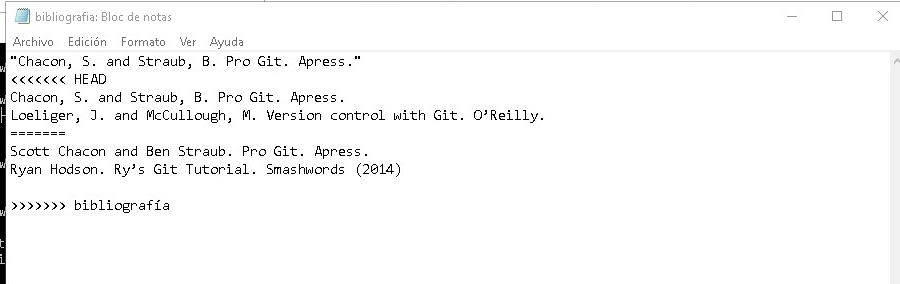
git commit –m “Añadida nueva referencia bibliográfica.”

1. **Fusionar la rama bibliografia con la rama master.**

Existen dos versiones de los ficheros una versión en cada una de las dos ramas. Nos dice que el merge automático no ha funcionado y que se arreglen las diferencias.



1. **Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Resuelto conflicto de bibliografía.”**

****

Resuelvo los conflictos, y se hace un git add nombre\_archivo. No usar git add . aunque se pueda.

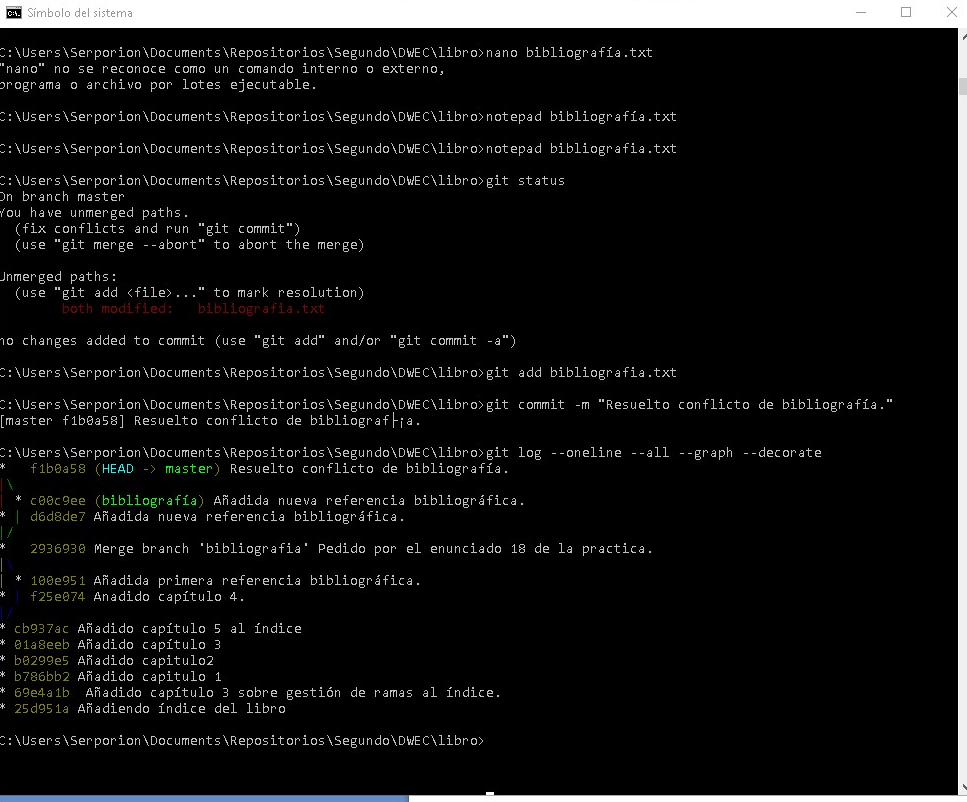
git add bibliografia.txt

git commit –m “Resuelto conflicto de bibliografía.”

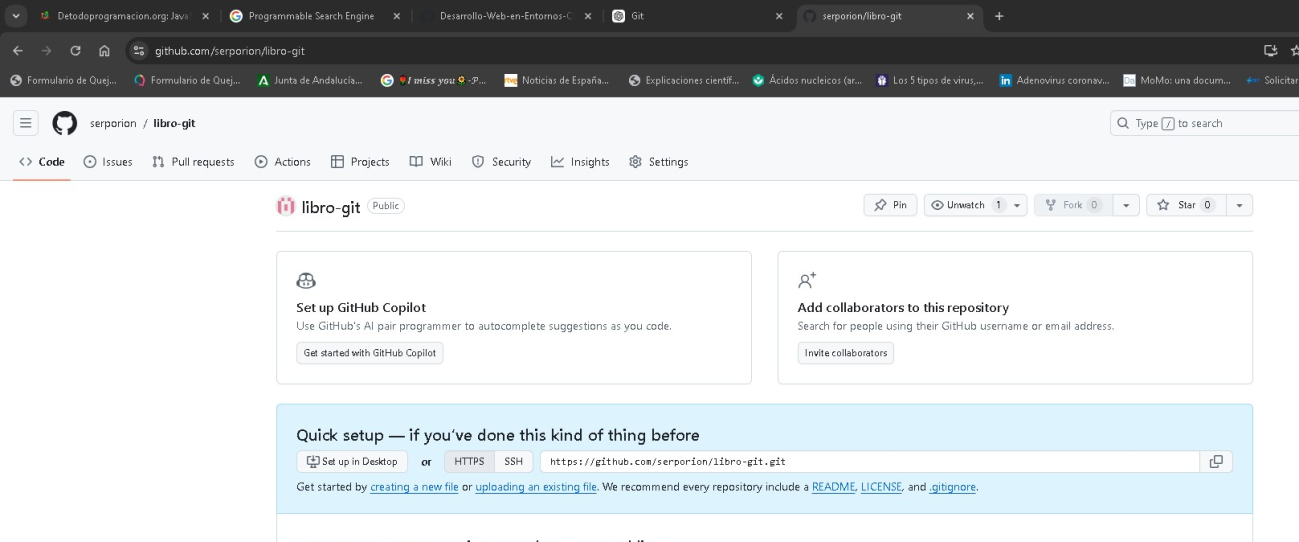
(El merge se hace después de ese commit. Está esperando a que ser resuelvan con dicho commit y automáticamente lo hace).

1. **Mostrar la historia del repositorio incluyendo todas las ramas.**

git log --oneline --all --graph –decorate

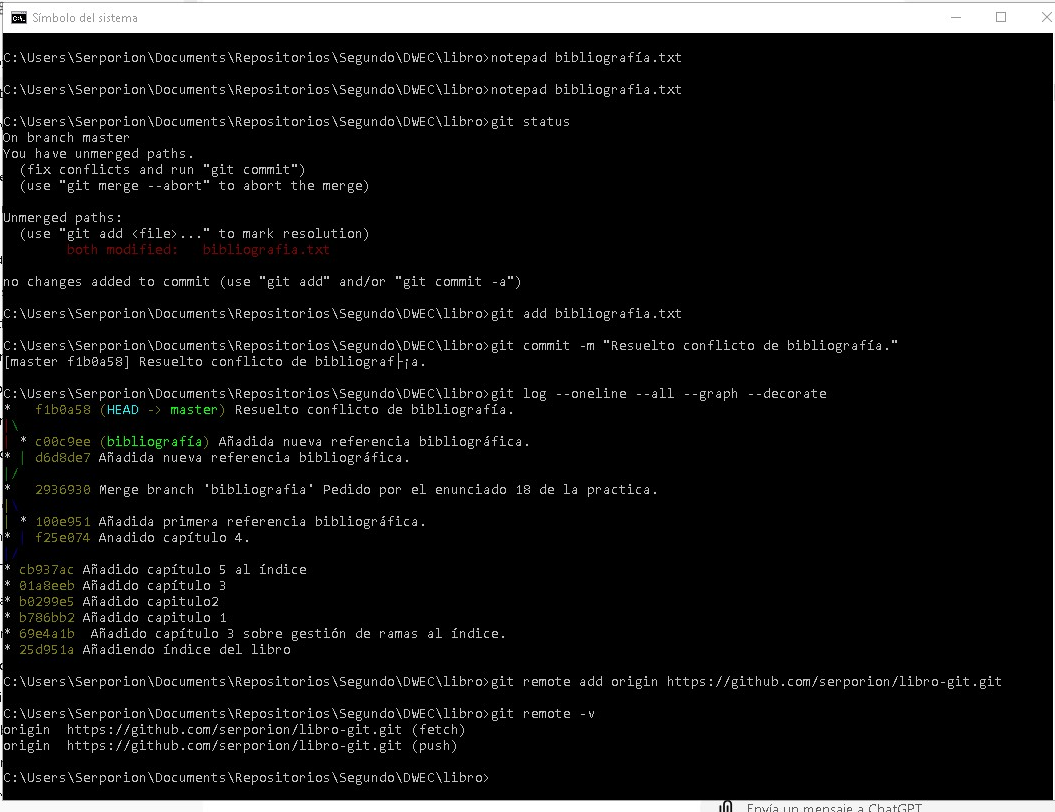


**Ejercicio 20**

1. **Crear un nuevo repositorio público en GitHub con el nombre libro-git.**
2. **Añadirlo al repositorio local del libro.**

git remote add origin https://github.com/serporion/libro-git.git

1. **Mostrar todos los repositorios remotos configurados.**

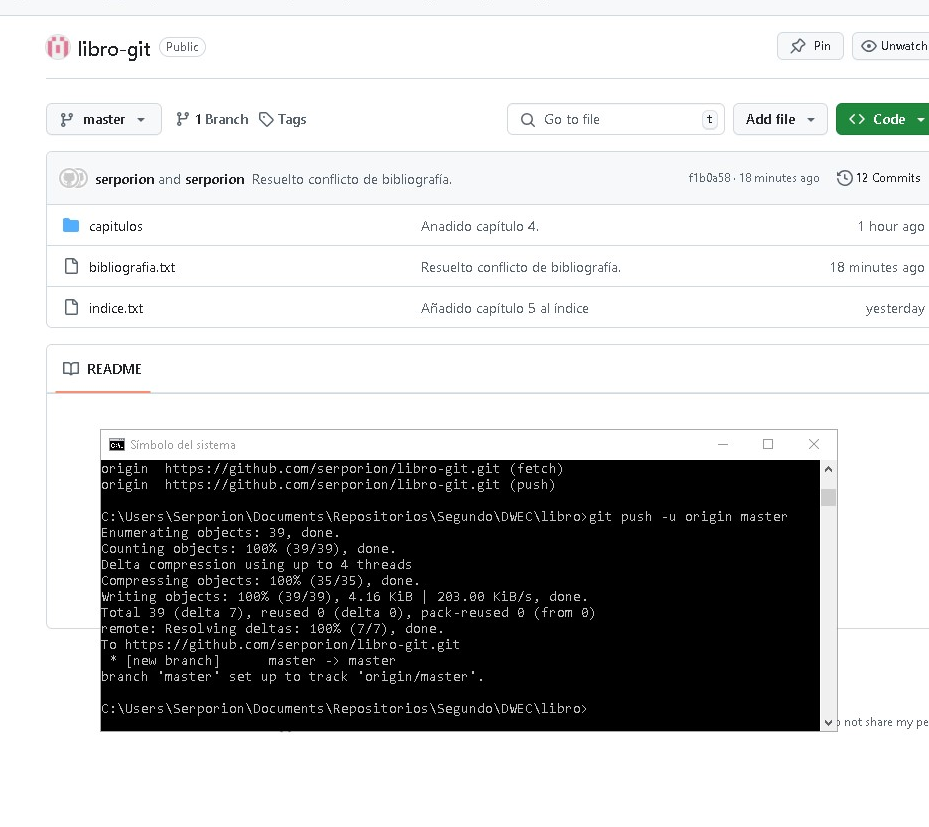
git remote -v

**Ejercicio 21**

1. **Añadir los cambios del repositorio local al repositorio remoto de GitHub.**

git push –u origin master

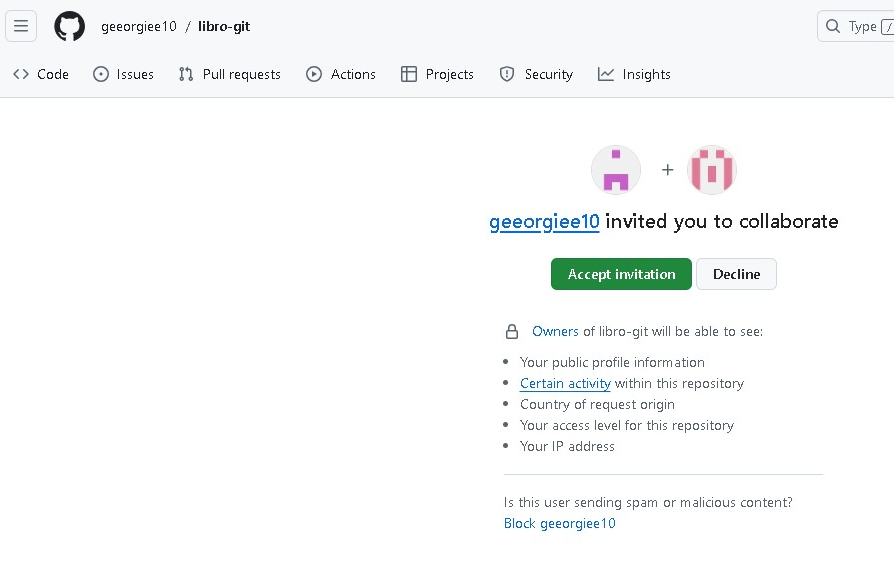
(como la rama bibliografía está fusionada con master, no me interesa subirla). Podría haber hecho: git push --all origin.

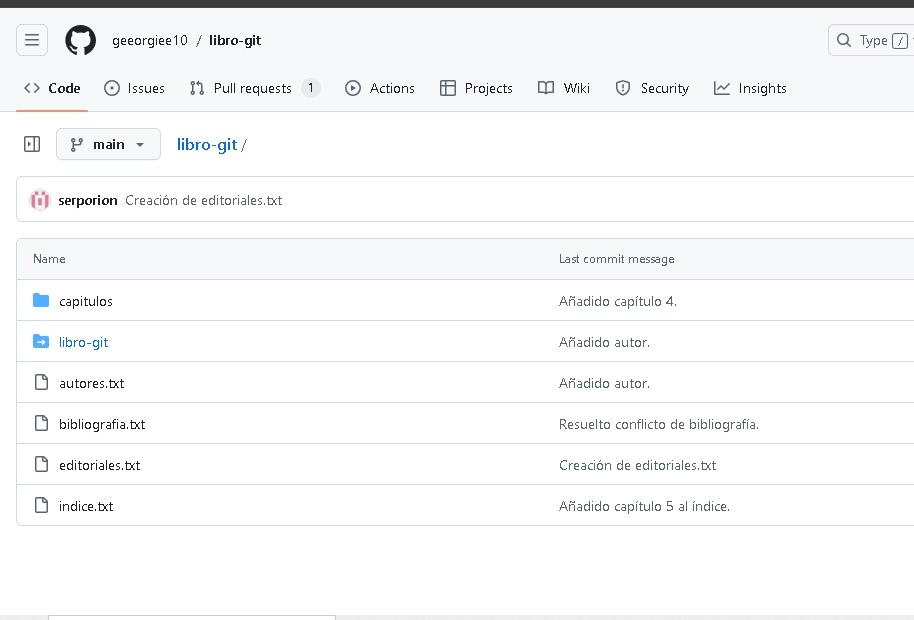
1. **Acceder a GitHub y comprobar que se han subido los cambios mostrando el historial de versiones.**

**Ejercicio 22**

1. **Colaborar en el repositorio remoto libro-git de otro usuario.**

Pedimos colaboración y mi compañero Jorge me acepta.





1. **Clonar su repositorio libro-git.**
2. **Añadir el fichero autores.txt que contenga el nombre del usuario y su correo electrónico.**

cd libro-git

notepad autores.txt

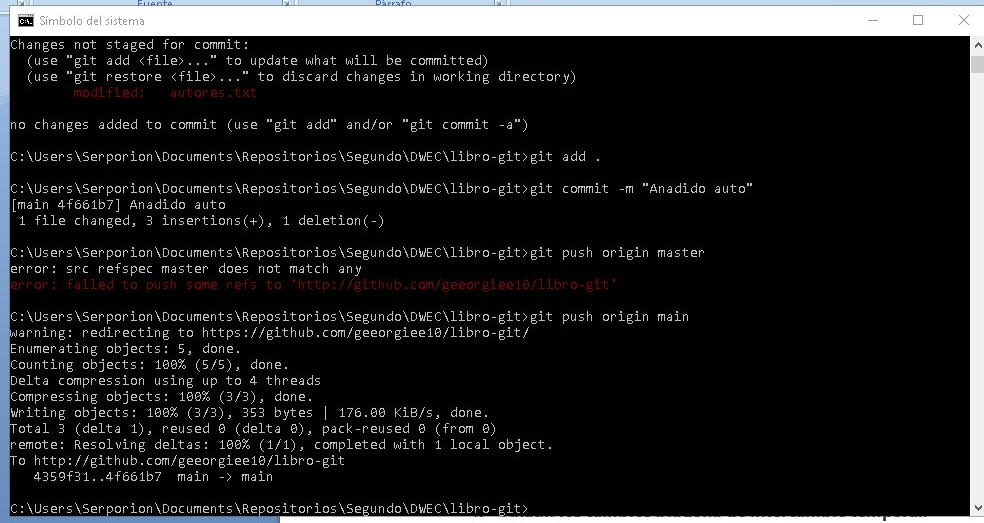
1. **Añadir los cambios a la zona de intercambio temporal.**

git add .

1. **Hacer un commit con el mensaje “Añadido autor.”**

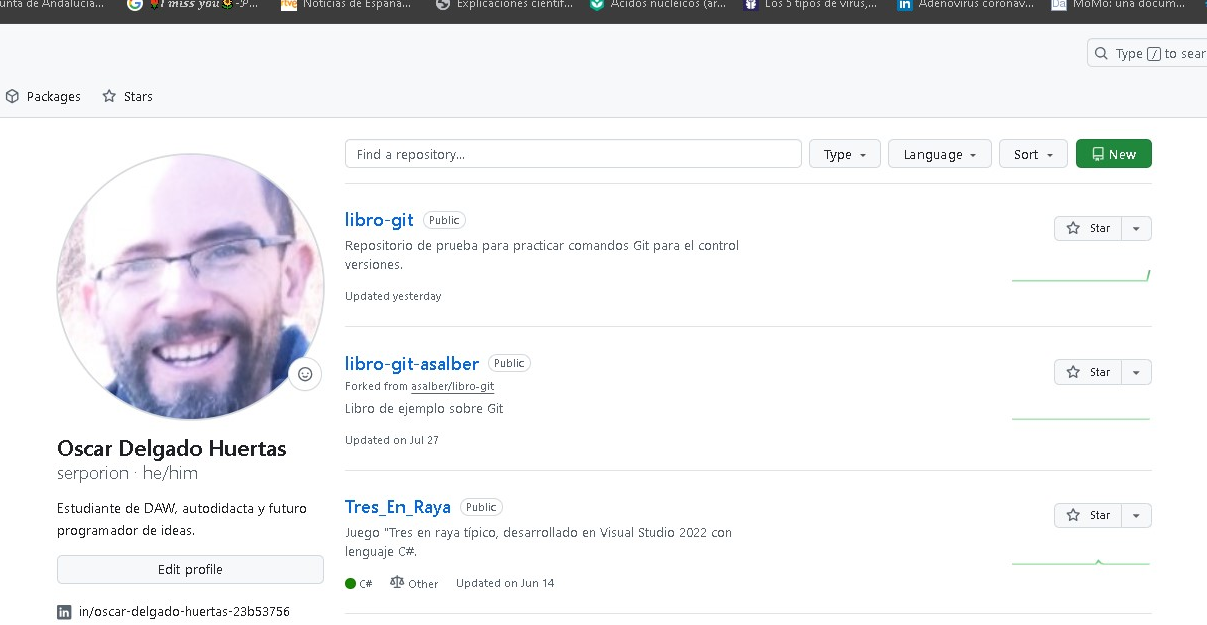
git commit –m “Añadido autor”

1. **Subir los cambios al repositorio remoto.**

git push origin master

**Ejercicio 23**

1. **Hacer una bifurcación del repositorio remoto asalber/libro-git en GitHub.**

****

1. **Clonar el repositorio creado en la cuenta de GitHub del usuario.**

git clone https://github.com/serporion/libro-git-asalber

1. **Crear una nueva rama autoria y activarla.**

git branch autoria

git checkout autoria

1. **Añadir el nombre del usuario y su correo al fichero autores.txt.**

notepad autores.txt

1. **Añadir los cambios a la zona de intercambio temporal.**

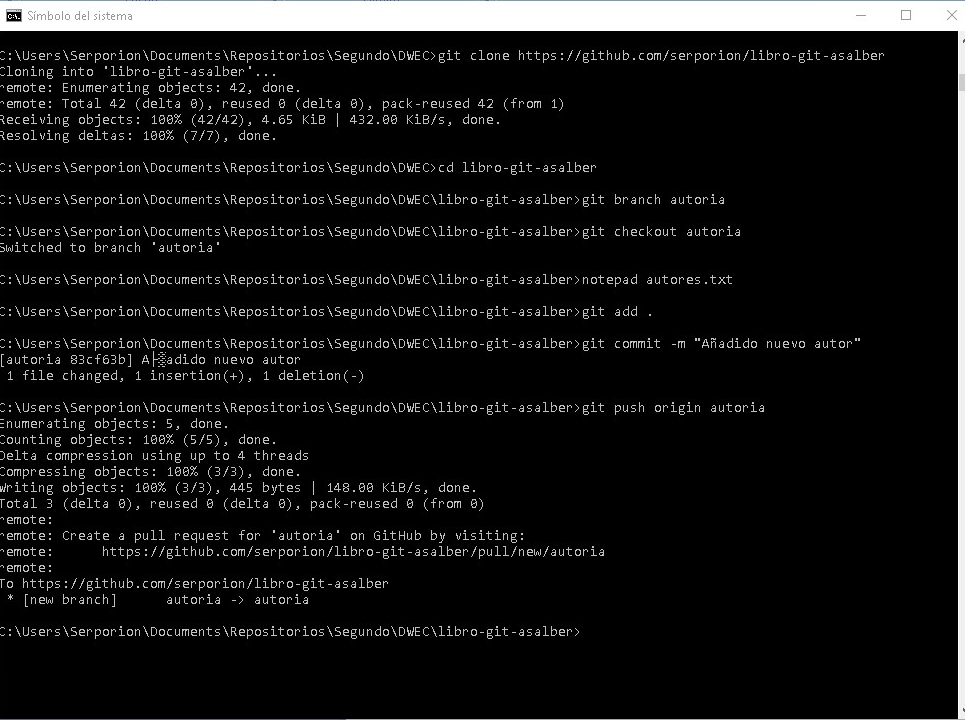
git add .

1. **Hacer un commit con el mensaje “Añadido nuevo autor.”**

git commit –m “Añadido nuevo autor”

1. **Subir los cambios de la rama autoria al repositorio remoto en GitHub.**

git push origin autoria



1. **Hacer un Pull Request de los cambios en la rama autoria.**

Comparamos el reposirio forkeado original y comparamos las ramas. Vemos que la rama autoria no existe en el repositorio base de “asalber”. Así que le damos a crear un pull request que posteriormente deberá ser revisado y aprobado por el propietario del repositorio.

