Git Git: Control de versiones

3º Evaluación: Línea de comandos.

Vamos a aprender las funcionalidades básicas para trabajar con el control de versiones GIT desde la línea de comandos.

[1 Ejercicios de creación y actualización de repositorios. 2](#_Toc66696880)

[2 Historial de cambios. 5](#_Toc66696881)

[3 Ejercicios de deshacer cambios 7](#_Toc66696882)

[4 Ejercicios de gestión de ramas 10](#_Toc66696883)

[5 Repositorios remotos 13](#_Toc66696884)

# Ejercicios de creación y actualización de repositorios.

## Ejercicio

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

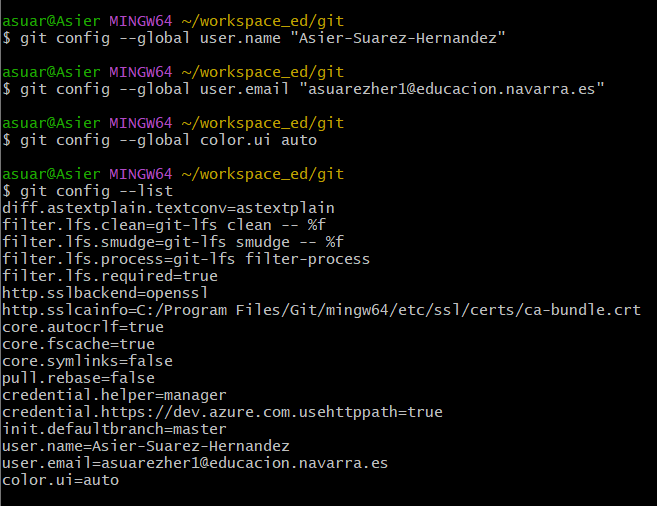
> git config --global user.name "Your-Full-Name"

> git config --global user.email "your-email-address"

> git config --global color.ui auto

> git config --list

Añade un pantallazo mostrando la configuración final.



## Ejercicio

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

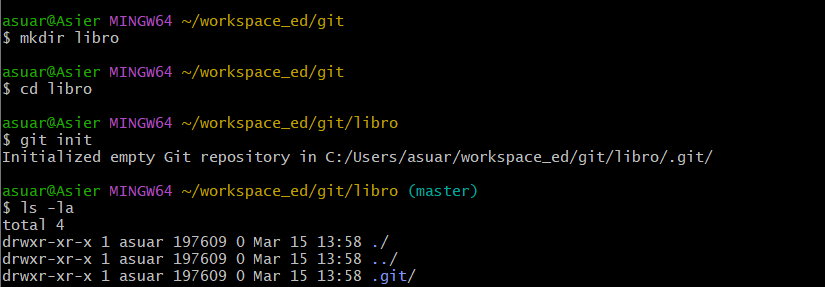
> mkdir libro

> cd libro

> git init

> ls -la

Añade un pantallazo mostrando la configuración final.



## Ejercicio

1. Comprobar el estado del repositorio.
2. Crear un fichero indice.txt con el siguiente contenido:

* Capítulo 1: Introducción a Git
* Capítulo 2: Flujo de trabajo básico
* Capítulo 3: Repositorios remotos

1. Comprobar de nuevo el estado del repositorio.
2. Añadir el fichero a la zona de intercambio temporal.
3. Volver a comprobar una vez más el estado del repositorio.

> git status

> cat > indice.txt

Capítulo 1: Introducción a Git

Capítulo 2: Flujo de trabajo básico

Capítulo 3: Repositorios remotos

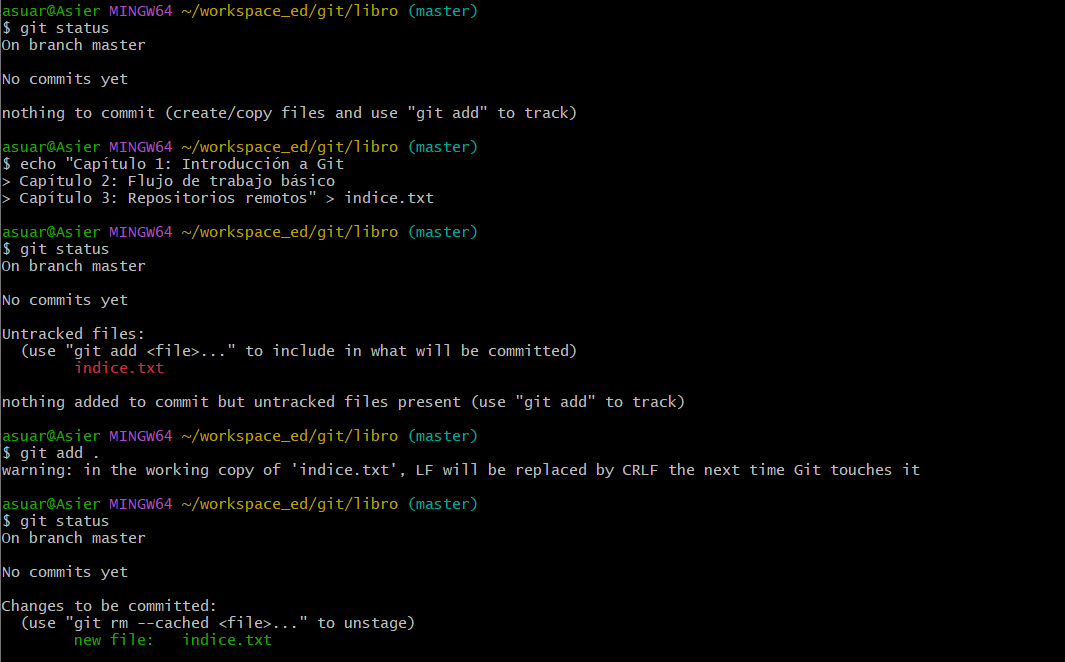
Ctrl+D

> git status

> git add indice.txt

> git status

Añade un pantallazo mostrando la ejecución de la secuencia de comandos.



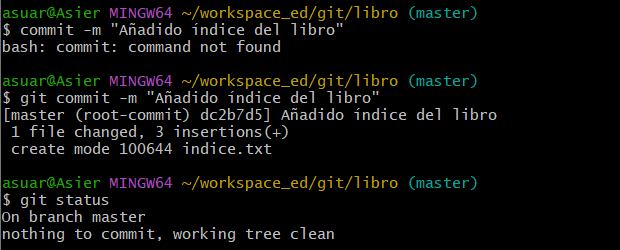
## Ejercicio

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

> git commit -m "Añadido índice del libro."

> git status

Añade un pantallazo mostrando la ejecución de la secuencia de comandos.



## Ejercicio

1. Cambiar el fichero indice.txt para que contenga lo siguiente:

* Capítulo 1: Introducción a Git
* Capítulo 2: Flujo de trabajo básico
* Capítulo 3: Gestión de ramas
* Capítulo 4: Repositorios remotos

1. Mostrar los cambios con respecto a la última versión guardada en el repositorio.
2. Hacer un commit de los cambios con el mensaje “Añadido capítulo 3 sobre gestión de ramas”.

> cat > indice.txt

Capítulo 1: Introducción a Git

Capítulo 2: Flujo de trabajo básico

Capítulo 3: Gestión de ramas

Capítulo 4: Repositorios remotos

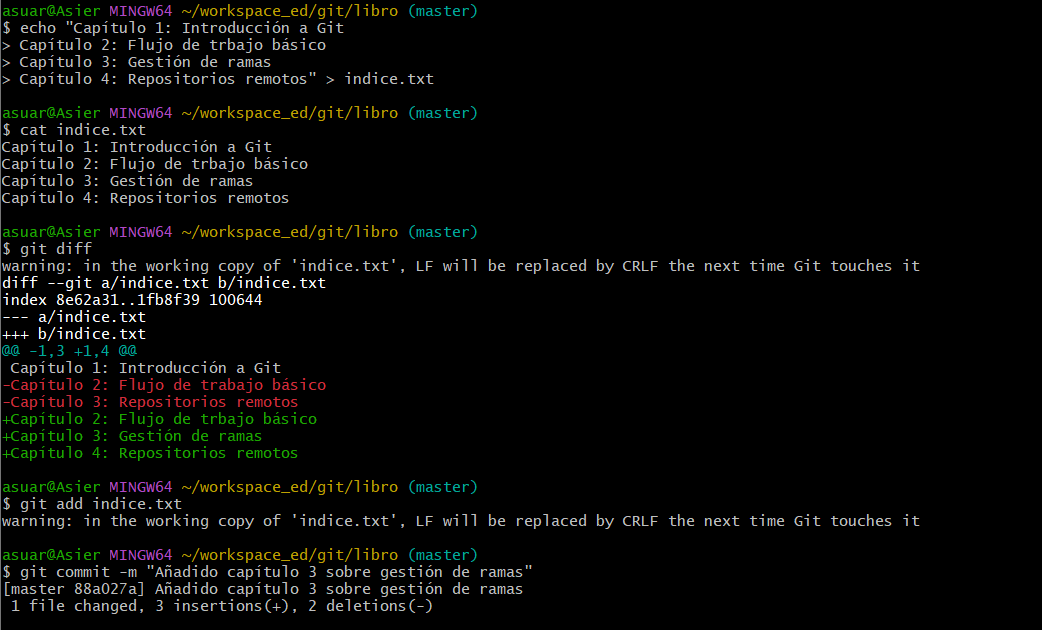
Ctrl+D

> git diff

> git add indice.txt

> git commit -m "Añadido capítulo 3 sobre gestión de ramas"

Añade un pantallazo mostrando la ejecución de la secuencia de comandos.



## Ejercicio

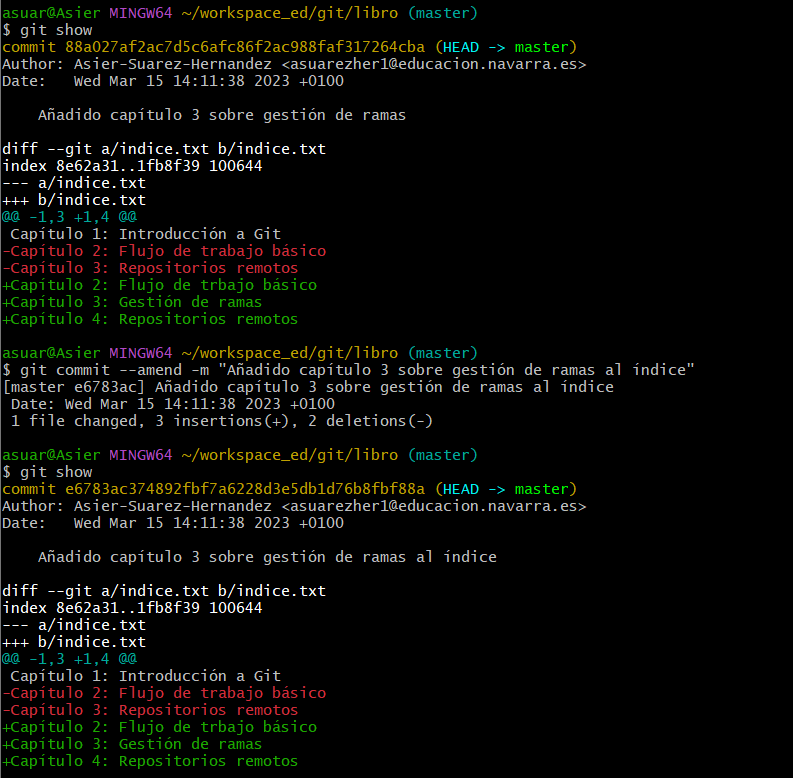
1. Mostrar los cambios de la última versión del repositorio con respecto a la anterior.
2. Cambiar el mensaje del último commit por “Añadido capítulo 3 sobre gestión de ramas al índice.”
3. Volver a mostrar los últimos cambios del repositorio.

> git show

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

git show

Añade un pantallazo mostrando la ejecución de la secuencia de comandos.



# Historial de cambios.

## Ejercicio

1. Mostrar el historial de cambios del repositorio.
2. Crear la carpeta capitulos y crear dentro de ella el fichero capitulo1.txt con el siguiente texto.

Git es un sistema de control de versiones ideado por Linus Torvalds.

1. Añadir los cambios a la zona de intercambio temporal.
2. Hacer un commit de los cambios con el mensaje “Añadido capítulo 1.”
3. Volver a mostrar el historial de cambios del repositorio.

> git log

> mkdir capitulos

> cat > capitulos/capitulo1.txt

Git es un sistema de control de versiones ideado por Linus Torvalds.

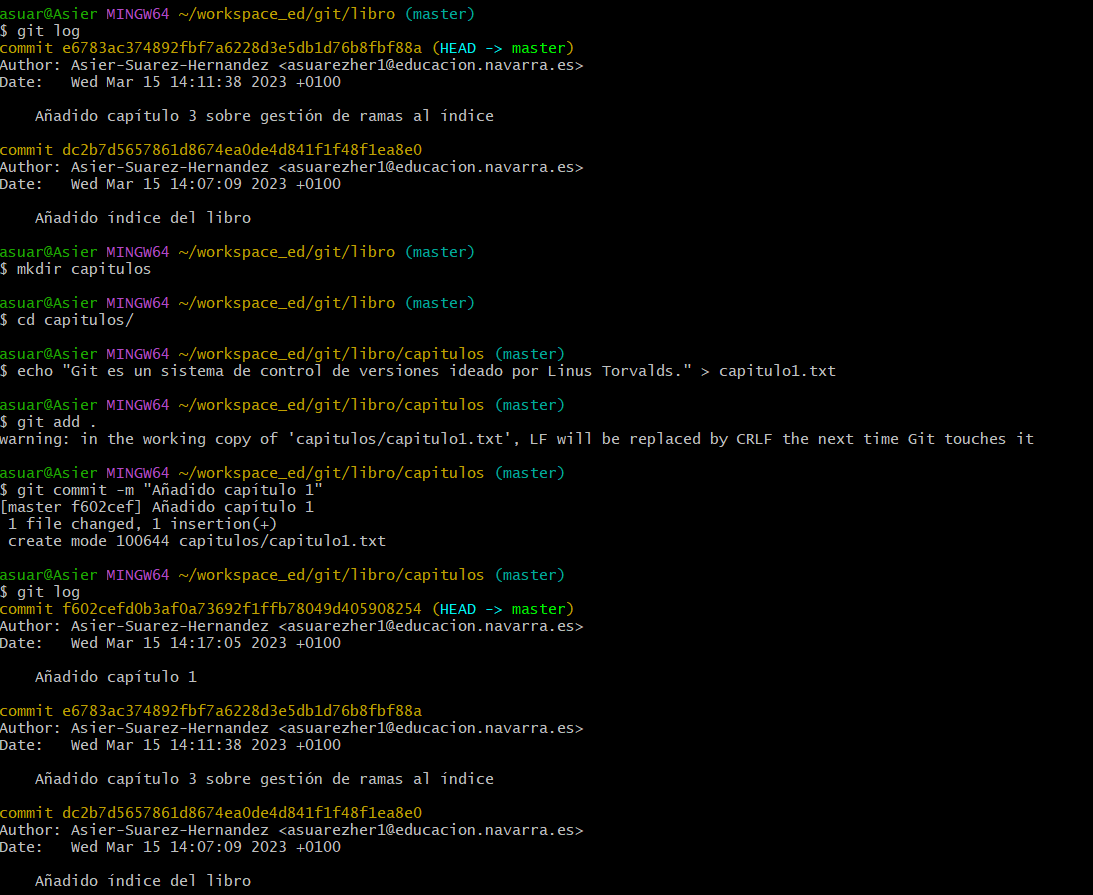
Ctrl+D

> git add .

> git commit -m "Añadido capítulo 1."

> git log

Añade un pantallazo mostrando la ejecución de la secuencia de comandos.



## Ejercicio

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

El flujo de trabajo básico con Git consiste en: 1- Hacer cambios en el repositorio. 2 - Añadir los cambios a la zona de intercambio temporal. 3- Hacer un commit de los cambios.

1. Añadir los cambios a la zona de intercambio temporal.
2. Hacer un commit de los cambios con el mensaje “Añadido capítulo 2.”
3. Mostrar las diferencias entre la última versión y dos versiones anteriores.

> cat > capitulos/capitulo2.txt

El flujo de trabajo básico con Git consiste en:

1- Hacer cambios en el repositorio.

2- Añadir los cambios a la zona de intercambio temporal.

3- Hacer un commit de los cambios.

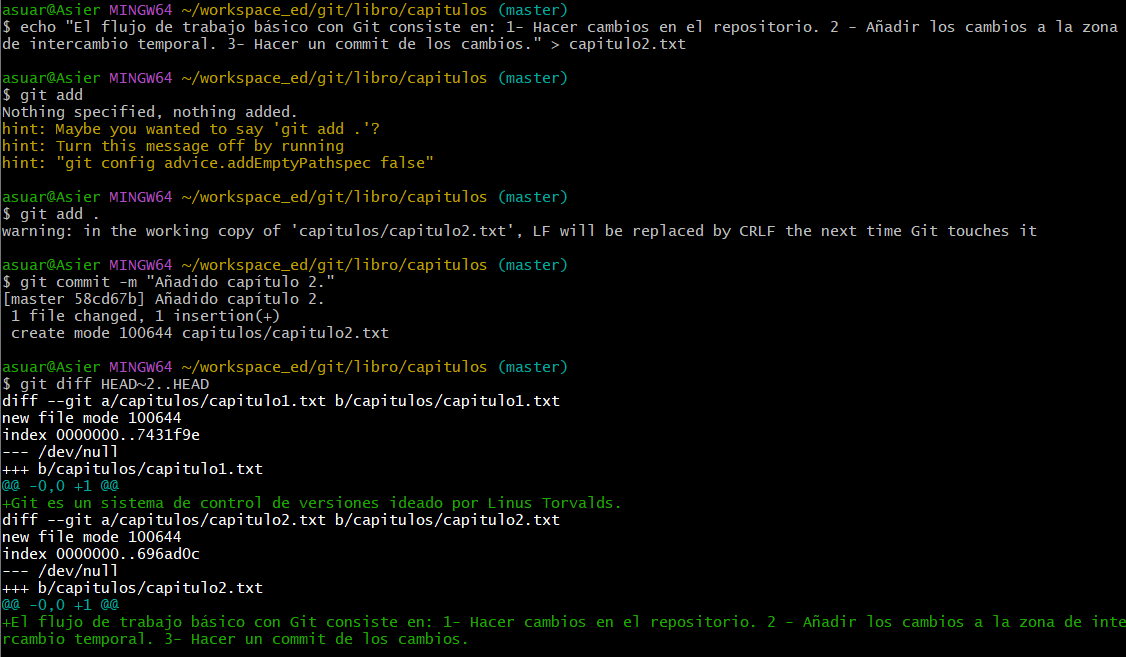
Ctrl+D

> git add .

> git commit -m "Añadido capítulo 2."

> git diff HEAD~2..HEAD

Añade un pantallazo mostrando la ejecución de la secuencia de comandos.



## Ejercicio

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

Git permite la creación de ramas lo que permite tener distintas versiones del mismo proyecto y trabajar de manera simultanea en ellas.

1. Añadir los cambios a la zona de intercambio temporal.
2. Hacer un commit de los cambios con el mensaje “Añadido capítulo 3.”
3. Mostrar las diferencias entre la primera y la última versión del repositorio.

> cat > capitulos/capitulo3.txt

Git permite la creación de ramas lo que permite tener distintas versiones del mismo proyecto y trabajar de manera simultanea en ellas.

Ctrl+D

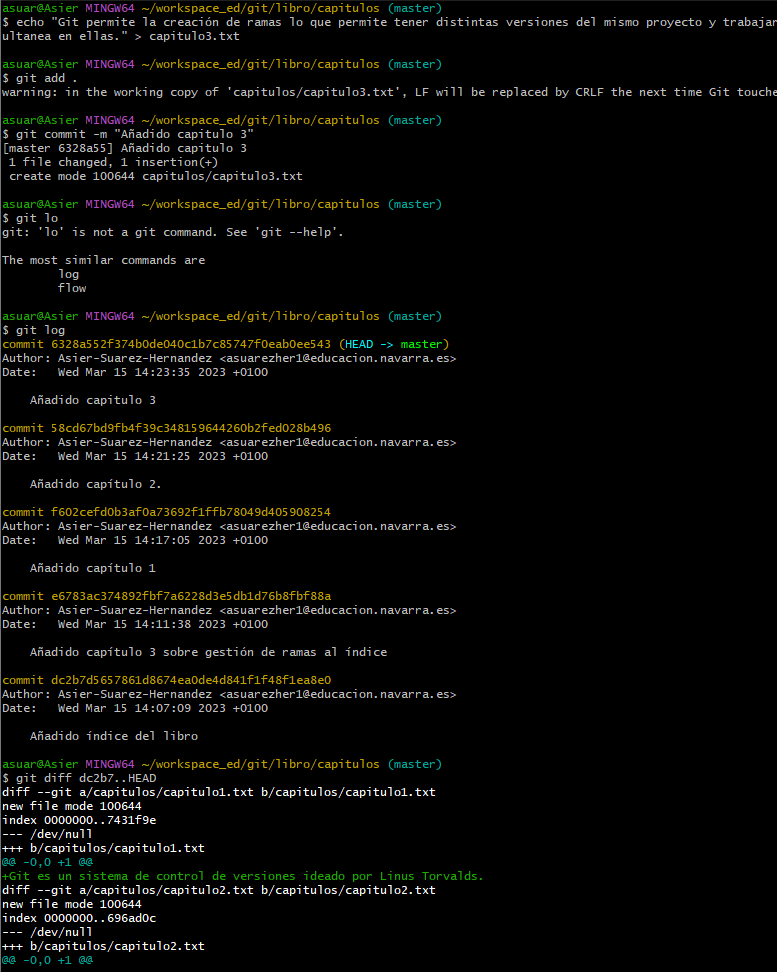
> git add .

> git commit -m "Añadido capítulo 3."

> git log

> git diff <codigo hash de la primera version>..HEAD

Añade un pantallazo mostrando la configuración final.



## Ejercicio

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

Capítulo 5: Conceptos avanzados

1. Añadir los cambios a la zona de intercambio temporal.
2. Hacer un commit de los cambios con el mensaje “Añadido capítulo 5 al índice.”.
3. Mostrar quién ha hecho cambios sobre el fichero indice.txt.

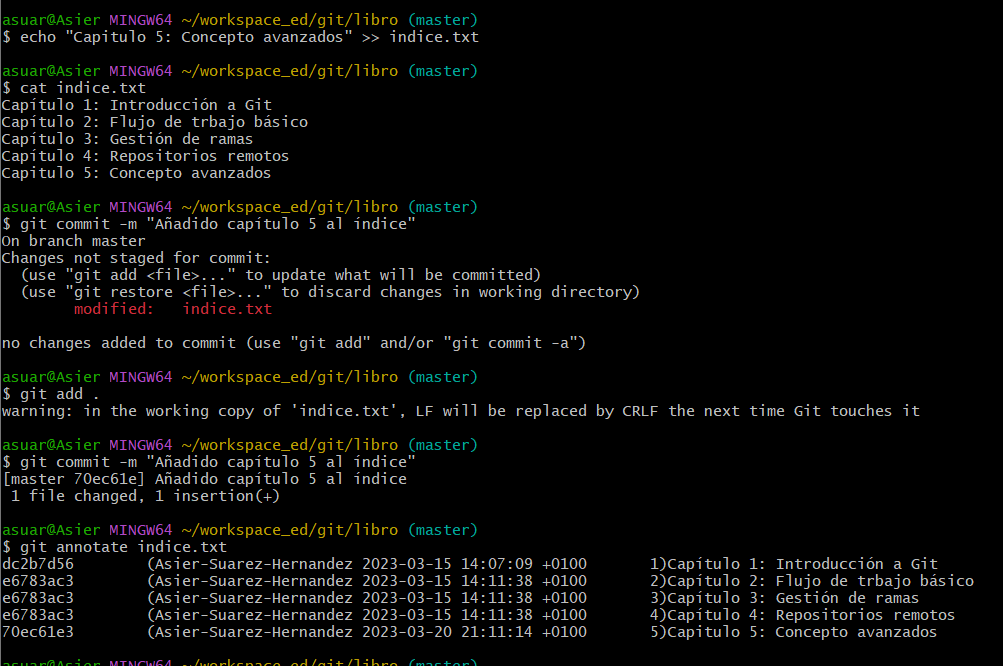
> echo "Capítulo 5: Conceptos avanzados" >> indice.txt

> git add .

> git commit -m "Añadido capítulo 5 al índice."

> git annotate indice.txt

Añade un pantallazo mostrando la configuración final.



# Ejercicios de deshacer cambios

## Ejercicio

1. Eliminar la última línea del fichero indice.txt y guardarlo.
2. Comprobar el estado del repositorio.
3. Deshacer los cambios realizados en el fichero indice.txt para volver a la versión anterior del fichero.
4. Volver a comprobar el estado del repositorio.

> nano indice.txt

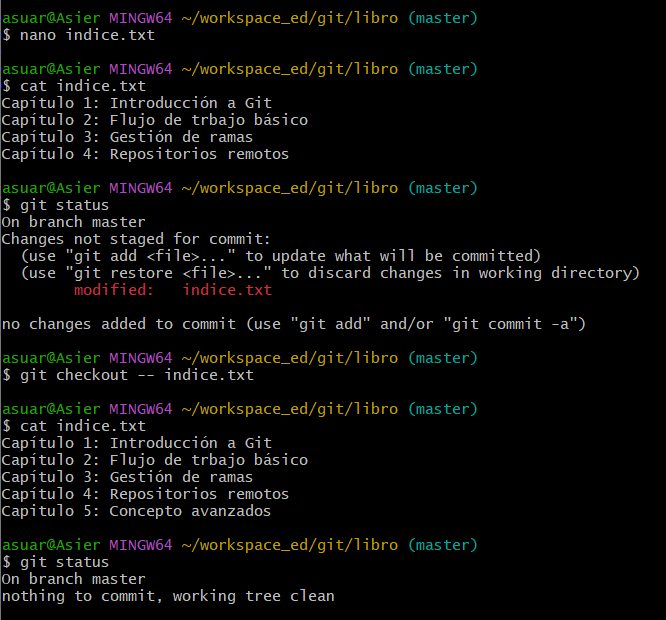
# Eliminar la última línea y guardar el fichero.

> git status

> git checkout -- indice.txt

> git status

Añade un pantallazo mostrando la configuración final.



## Ejercicio

1. Eliminar la última línea del fichero indice.txt y guardarlo.
2. Añadir los cambios a la zona de intercambio temporal.
3. Comprobar de nuevo el estado del repositorio.
4. Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.
5. Comprobar de nuevo el estado del repositorio.
6. Deshacer los cambios realizados en el fichero indice.txt para volver a la versión anterior del fichero.
7. Volver a comprobar el estado del repositorio.

> nano indice.txt

# Eliminar la última línea y guardar el fichero.

> git add .

> git status

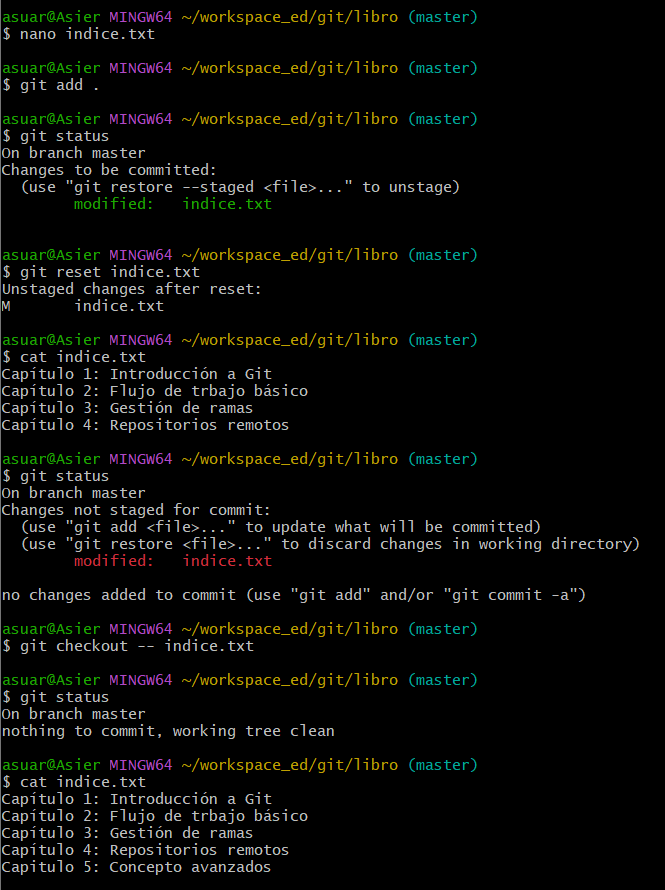
> git reset indice.txt

> git status

> git checkout -- indice.txt

> git status

Añade un pantallazo mostrando la configuración final.



## Ejercicio

1. Eliminar la última línea del fichero indice.txt y guardarlo.
2. Eliminar el fichero capitulos/capitulo3.txt.
3. Añadir un fichero nuevo captitulos/capitulo4.txt vacío.
4. Añadir los cambios a la zona de intercambio temporal.
5. Comprobar de nuevo el estado del repositorio.
6. Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.
7. Comprobar de nuevo el estado del repositorio.
8. Deshacer los cambios realizados para volver a la versión del repositorio.
9. Volver a comprobar el estado del repositorio.

> nano indice.txt

# Eliminar la última línea y guardar el fichero.

> rm capitulos/capitulo3.txt

> touch capitulos/capitulo4.txt

> git add .

> git status

> git reset

> git status

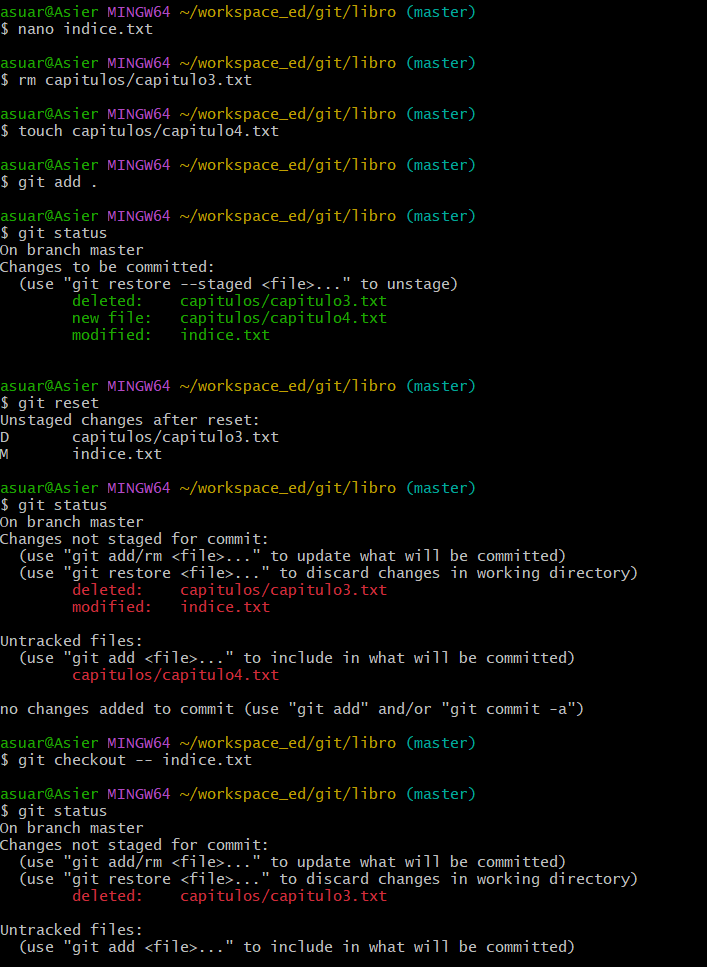
> git checkout -- .

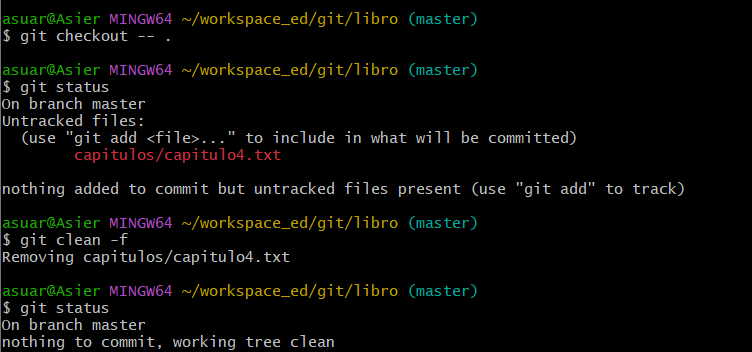
> git status

> git clean -f

> git status

Añade un pantallazo mostrando la configuración final.





## Ejercicio

1. Eliminar la última línea del fichero indice.txt y guardarlo.
2. Eliminar el fichero capitulos/capitulo3.txt.
3. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Borrado accidental.”
4. Comprobar el historial del repositorio.
5. Deshacer el último commit pero mantener los cambios anteriores en el directorio de trabajo y la zona de intercambio temporal.
6. Comprobar el historial y el estado del repositorio.
7. Volver a hacer el commit con el mismo mensaje de antes.
8. Deshacer el último commit y los cambios anteriores del directorio de trabajo volviendo a la versión anterior del repositorio.
9. Comprobar de nuevo el historial y el estado del repositorio.

> nano indice.txt

# Eliminar la última línea y guardar el fichero.

> rm capitulos/capitulo3.txt

> git commit -am "Borrado accidental."

> git status

> git log

> git reset --soft HEAD~1

> git status

> git commit -m "Borrado accidental."

> git status

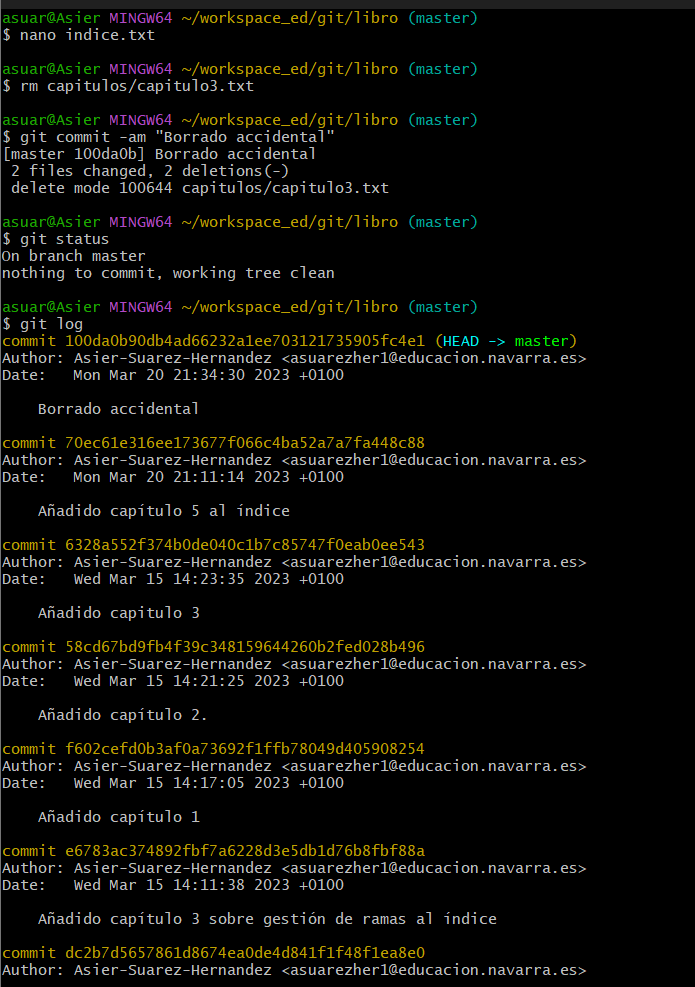
> git log

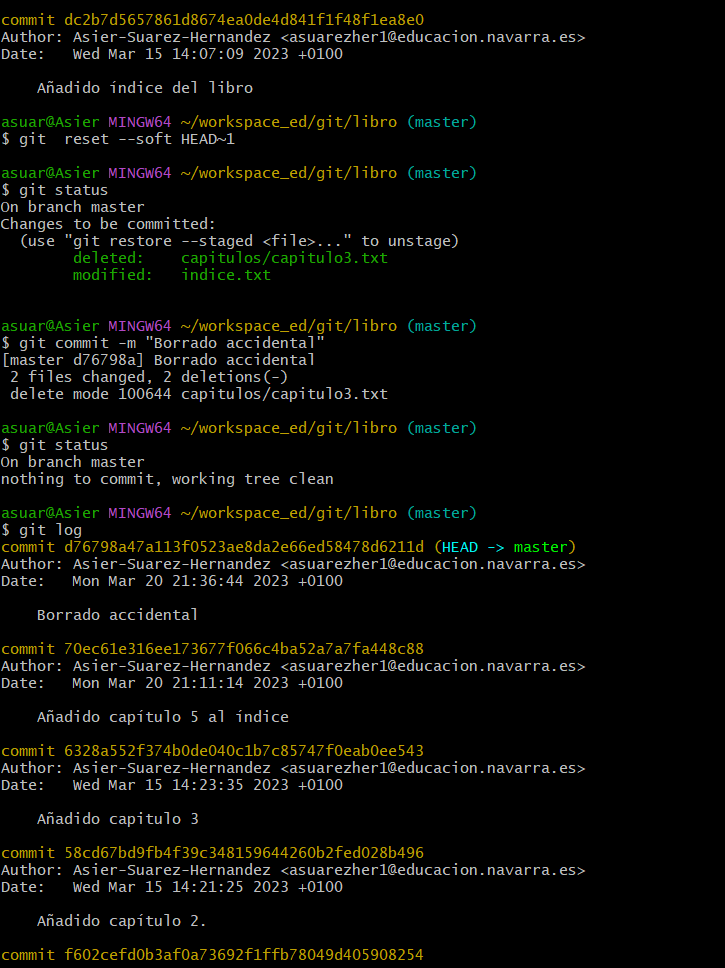
> git reset --hard HEAD~1

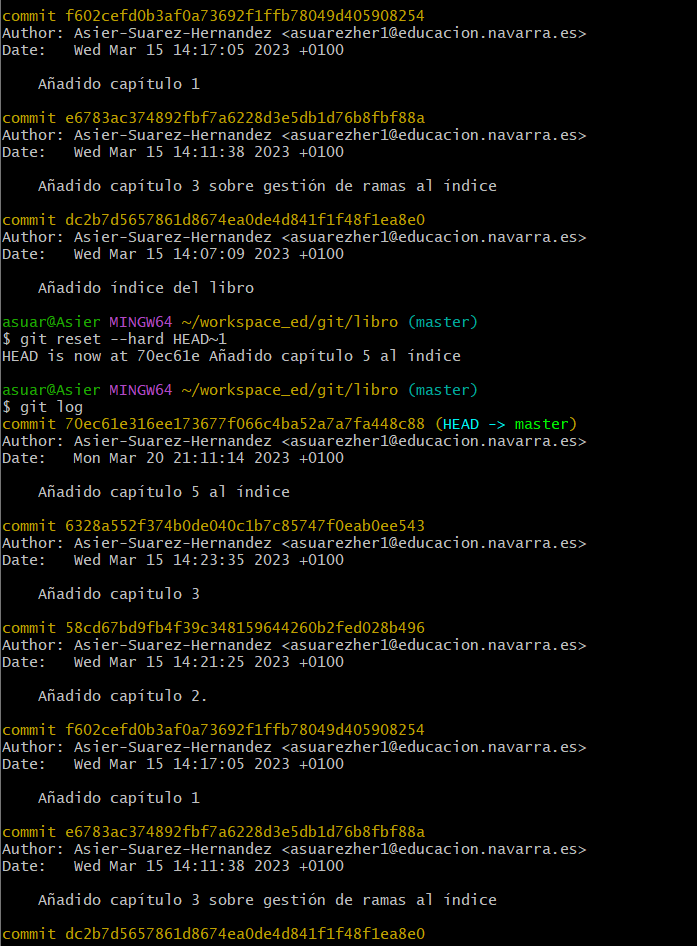
> git log

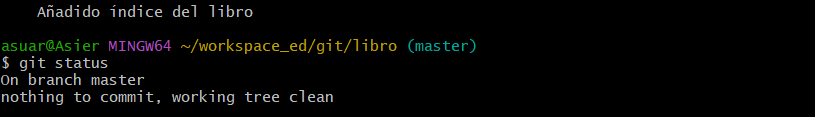
> git status

Añade un pantallazo mostrando la configuración final.









# Ejercicios de gestión de ramas

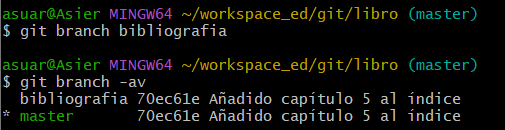
## Ejercicio

Crear una nueva rama bibliografia y mostrar las ramas del repositorio.

> git branch bibliografia

> git branch -av

Añade un pantallazo mostrando la configuración final.



## Ejercicio

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

En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto.

1. Añadir los cambios a la zona de intercambio temporal.
2. Hacer un commit con el mensaje “Añadido capítulo 4.”
3. Mostrar la historia del repositorio incluyendo todas las ramas.

> cat > capitulos/capitulo4.txt

En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto.

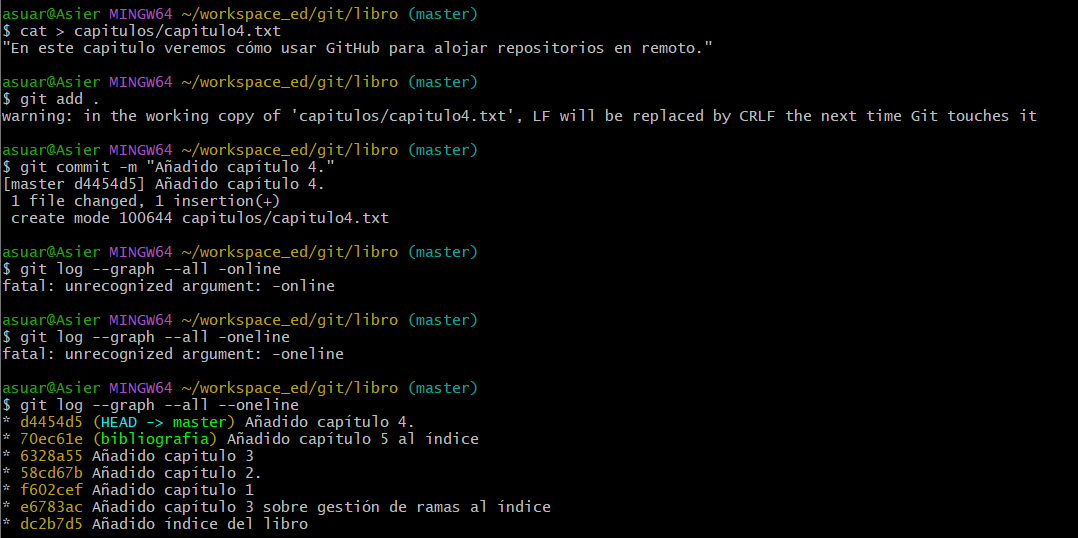
Ctrl+D

> git add .

> git commit -m "Añadido capítulo 4."

> git log --graph --all –oneline

Añade un pantallazo mostrando la configuración final.



## Ejercicio

1. Cambiar a la rama bibliografia.
2. Crear el fichero bibliografia.txt y añadir la siguiente referencia

Chacon, S. and Straub, B. Pro Git. Apress.

1. Añadir los cambios a la zona de intercambio temporal.
2. Hacer un commit con el mensaje “Añadida primera referencia bibliográfica.”
3. Mostrar la historia del repositorio incluyendo todas las ramas.

> git checkout bibliografia

> cat > bibliografia.txt

- Chacon, S. and Straub, B. Pro Git. Apress.

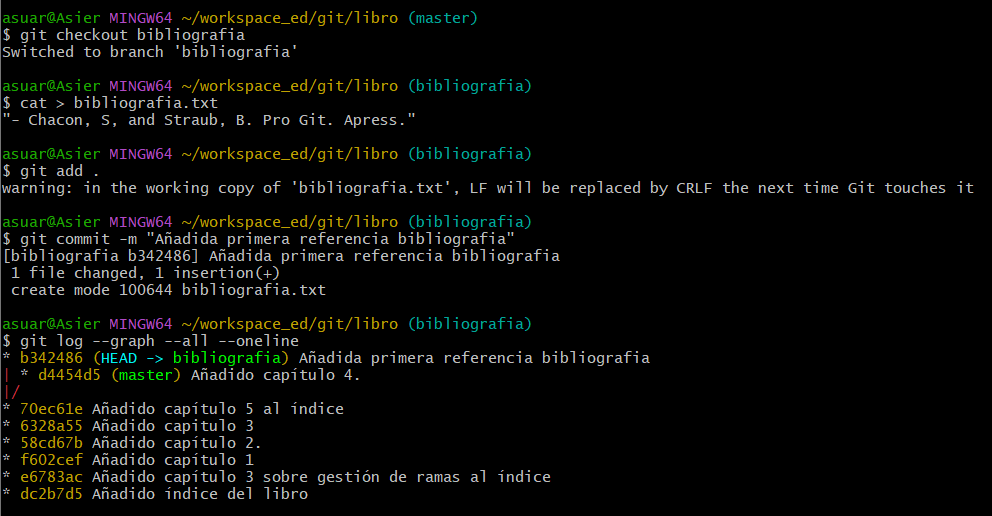
Ctrl+D

> git add .

> git commit -m "Añadida primera referencia bibliográfica."

> git log --graph --all –oneline

Añade un pantallazo mostrando la configuración final.



## Ejercicio

1. Fusionar la rama bibliografia con la rama master.
2. Mostrar la historia del repositorio incluyendo todas las ramas.
3. Eliminar la rama bibliografia.
4. Mostrar de nuevo la historia del repositorio incluyendo todas las ramas.

> git checkout master

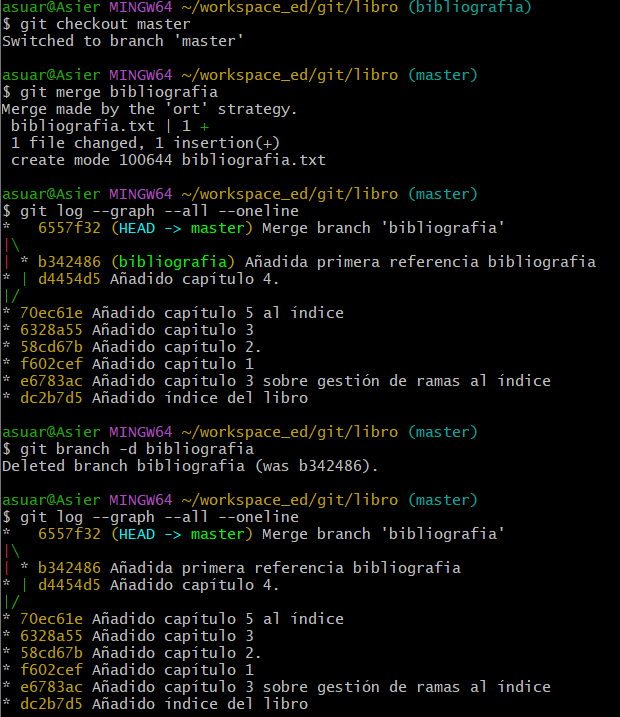
> git merge bibliografia

> git log --graph --all --oneline

> git branch -d bibliografia

> git log --graph --all –oneline

Añade un pantallazo mostrando la configuración final.



## Ejercicio

1. Crear la rama bibliografia.
2. Cambiar a la rama bibliografia.
3. Cambiar el fichero bibliografia.txt para que contenga las siguientes referencias:

Scott Chacon and Ben Straub. Pro Git. Apress.

Ryan Hodson. Ry’s Git Tutorial. Smashwords (2014)

1. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Añadida nueva referencia bibliográfica.”
2. Cambiar a la rama master.
3. Cambiar el fichero bibliografia.txt para que contenga las siguientes referencias:

Chacon, S. and Straub, B. Pro Git. Apress.

Loeliger, J. and McCullough, M. Version control with Git. O’Reilly.

1. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Añadida nueva referencia bibliográfica.”
2. Fusionar la rama bibliografia con la rama master.
3. Resolver el conflicto dejando el fichero bibliografia.txt con las referencias:

Chacon, S. and Straub, B. Pro Git. Apress.

Loeliger, J. and McCullough, M. Version control with Git. O’Reilly.

Hodson, R. Ry’s Git Tutorial. Smashwords (2014)

1. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Resuelto conflicto de bibliografía.”
2. Mostrar la historia del repositorio incluyendo todas las ramas.

> git branch bibliografia

> git checkout bibliografia

> cat > bibliografia.txt

- Scott Chacon and Ben Straub. Pro Git. Apress.

- Ryan Hodson. Ry's Git Tutorial. Smashwords (2014)

Ctrl+D

> git commit -a -m "Añadida nueva referencia bibliográfica."

> git checkout master

> cat > bibliografia.txt

- Chacon, S. and Straub, B. Pro Git. Apress.

- Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.

Ctrl+D

> git commit -a -m "Añadida nueva referencia bibliográfica."

> git merge bibliografia

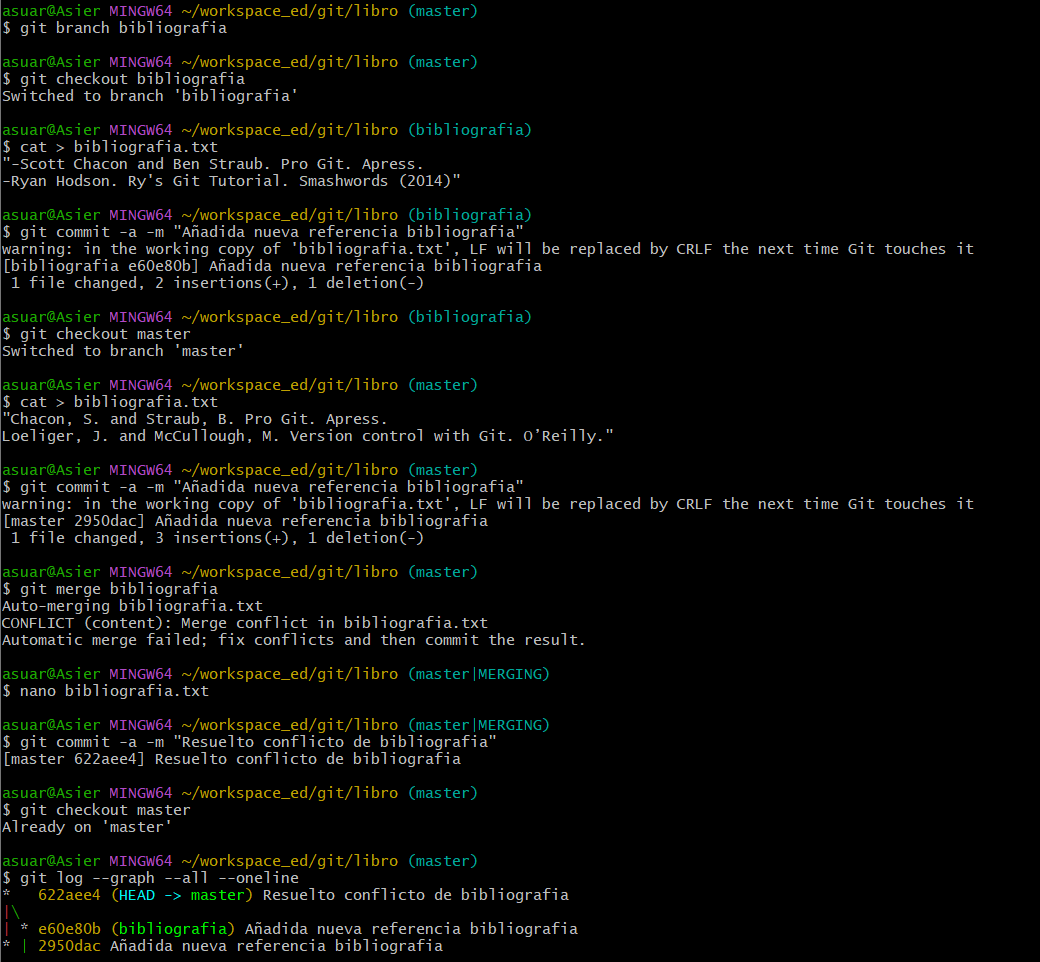
**> nano bibliografia.txt**

# Hacer los cambios indicados en el fichero

> git commit -a -m "Solucionado conflicto bibliografía."

> git log --graph --all **--oneline**

Añade un pantallazo mostrando la configuración final.



# Repositorios remotos

## Ejercicio

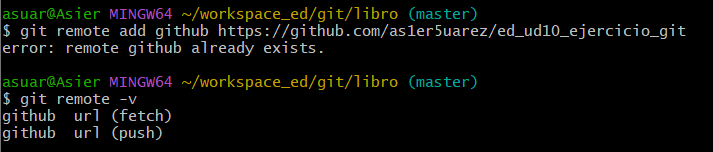
1. Crear un nuevo repositorio público en GitHub con el nombre libro-git.
2. Añadirlo al repositorio local del libro.
3. Mostrar todos los repositorios remotos configurados.

# Crear el repositorio en GitHub y copiar su url con protocolo https.

> git remote add github url

> git remote -v

Añade un pantallazo mostrando la configuración final.

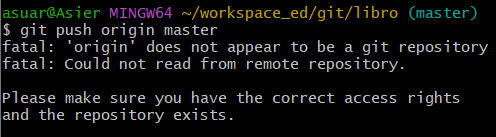


## Ejercicio

1. Añadir los cambios del repositorio local al repositorio remoto de GitHub.
2. Acceder a GitHub y comprobar que se han subido los cambios mostrando el historial de versiones.

>git push origin master

Añade un pantallazo mostrando la configuración final.



## Ejercicio

1. Colaborar en el repositorio remoto libro-git de otro usuario.
2. Clonar su repositorio libro-git.
3. Añadir el fichero autores.txt que contenga el nombre del usuario y su correo electrónico.
4. Añadir los cambios a la zona de intercambio temporal.
5. Hacer un commit con el mensaje “Añadido autor.”
6. Subir los cambios al repositorio remoto.

# Entrar en GigHub en el proyecto libro-git del que seamos colaboradores y copiar la url.

> git clone url

> cat > autores.txt

# Escribir el nombre del autor y su correo.

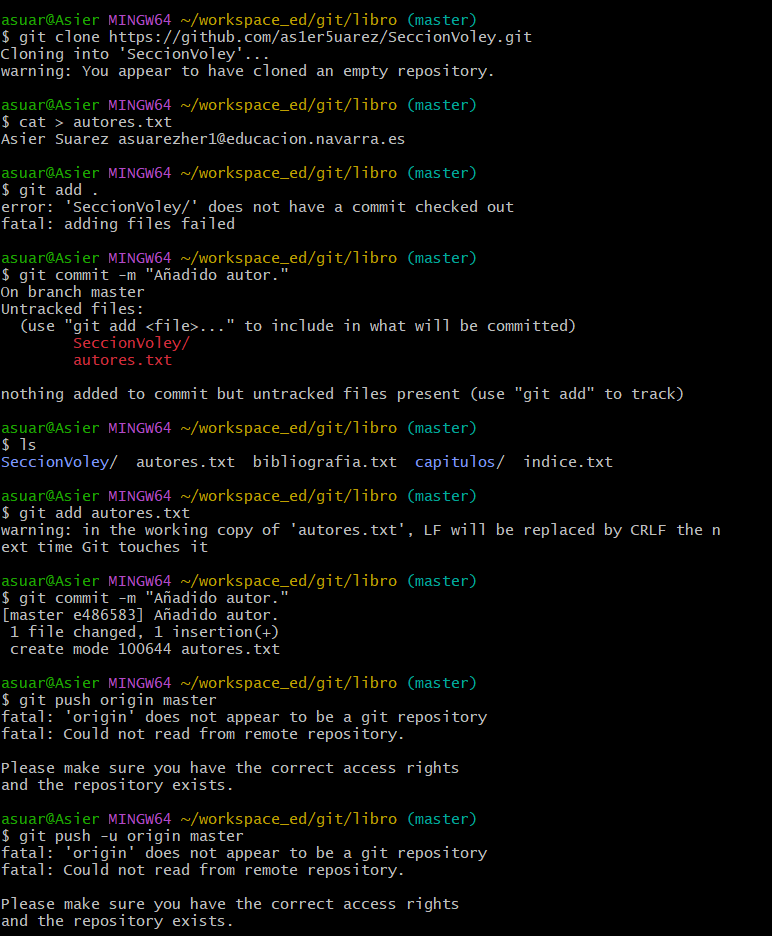
Ctrl+D

> git add .

> git commit -m "Añadido autor."

> git push origin master.

Añade un pantallazo mostrando la configuración final.



## Ejercicio

1. Hacer una bifurcación del repositorio remoto asalber/libro-git en GitHub.
2. Clonar el repositorio creado en la cuenta de GitHub del usuario.
3. Crear una nueva rama autoria y activarla.
4. Añadir el nombre del usuario y su correo al fichero autores.txt.
5. Añadir los cambios a la zona de intercambio temporal.
6. Hacer un commit con el mensaje “Añadido nuevo autor.”
7. Subir los cambios de la rama autoria al repositorio remoto en GitHub.
8. Hacer un Pull Request de los cambios en la rama autoria.

# Hacer el fork del repositorio asalber/libro-git en GitHub y copiar la url del repositorio creado en la cuenta de GitHub del usuario.

> git clone url

> git checkout -b autoria

# Editar con nano el fichero autores.txt y añadir el nombre y el correo electrónico del usuario en una nueva línea.

> git commit -am "Añadido nuevo autor."

> git push origin autoria

# Ir al repositorio remoto en GitHub y hacer clic en el botón Compare & Pull Request y después completar la solicitud haciendo clic en el botón Create Pull Request.

Añade un pantallazo mostrando la configuración final.

