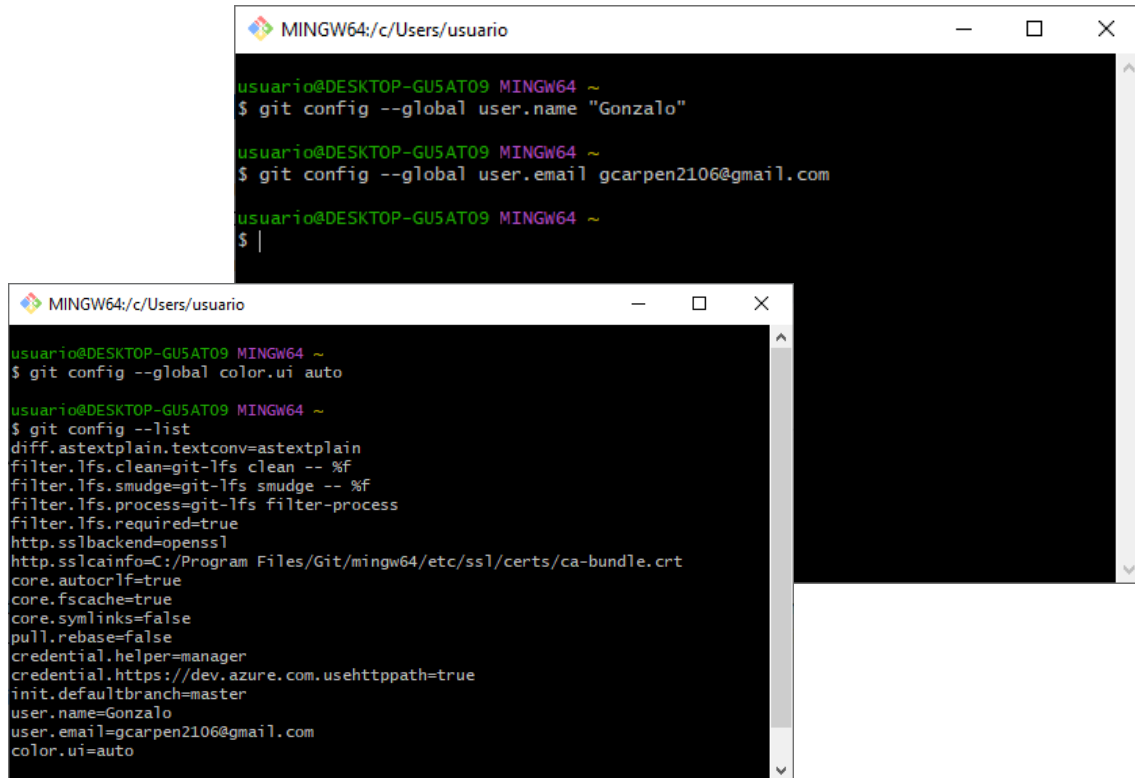


1. Instalar Git en vuestro sistema. Una vez instalado, si utilizáis Windows utilizar el emulador Git Bash para configurarlo con vuestro nombre de usuario, vuestro correo electrónico g.educaand.es. Git puede marcar con colores los resultados que muestra en tu terminal, ayudándote así a leerlos más fácilmente., activar también el coloreado de la salida. Finalmente Mostrar la configuración final.



```
MINGW64:/c/Users/usuario

usuario@DESKTOP-GU5AT09 MINGW64 ~
$ git config --global user.name "Gonzalo"

usuario@DESKTOP-GU5AT09 MINGW64 ~
$ git config --global user.email gcarpen2106@gmail.com

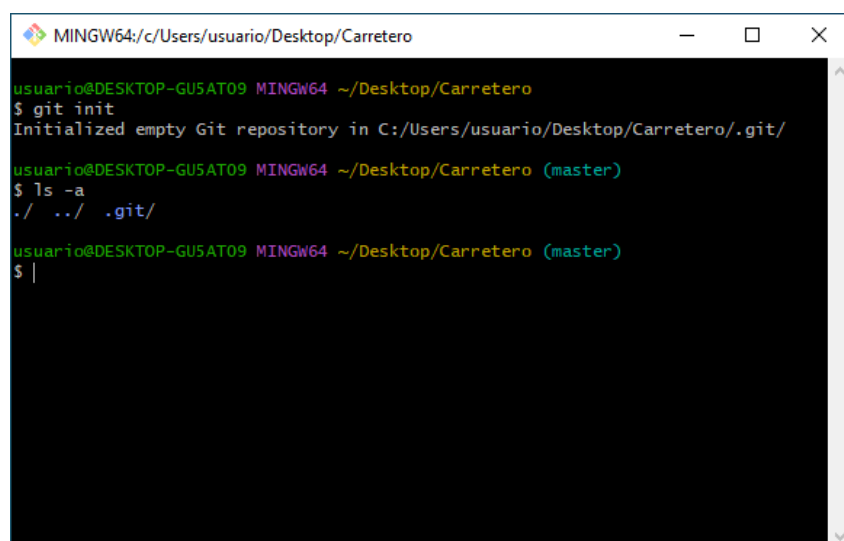
usuario@DESKTOP-GU5AT09 MINGW64 ~
$ |

MINGW64:/c/Users/usuario

usuario@DESKTOP-GU5AT09 MINGW64 ~
$ git config --global color.ui auto

usuario@DESKTOP-GU5AT09 MINGW64 ~
$ 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/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Gonzalo
user.email=gcarpen2106@gmail.com
color.ui=auto
```

2. Crear un nuevo repositorio local con el nombre "primerapellido" y mostrar su contenido. Ten en cuenta que cuando abres el emulador Git Bash, de forma predeterminada está trabajando en el home del usuario que instaló la aplicación Git.



```
MINGW64:/c/Users/usuario/Desktop/Carretero

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero
$ git init
Initialized empty Git repository in C:/Users/usuario/Desktop/Carretero/.git/

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ ls -a
./ ../ .git/

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |
```

3. Comprobar el estado del repositorio.

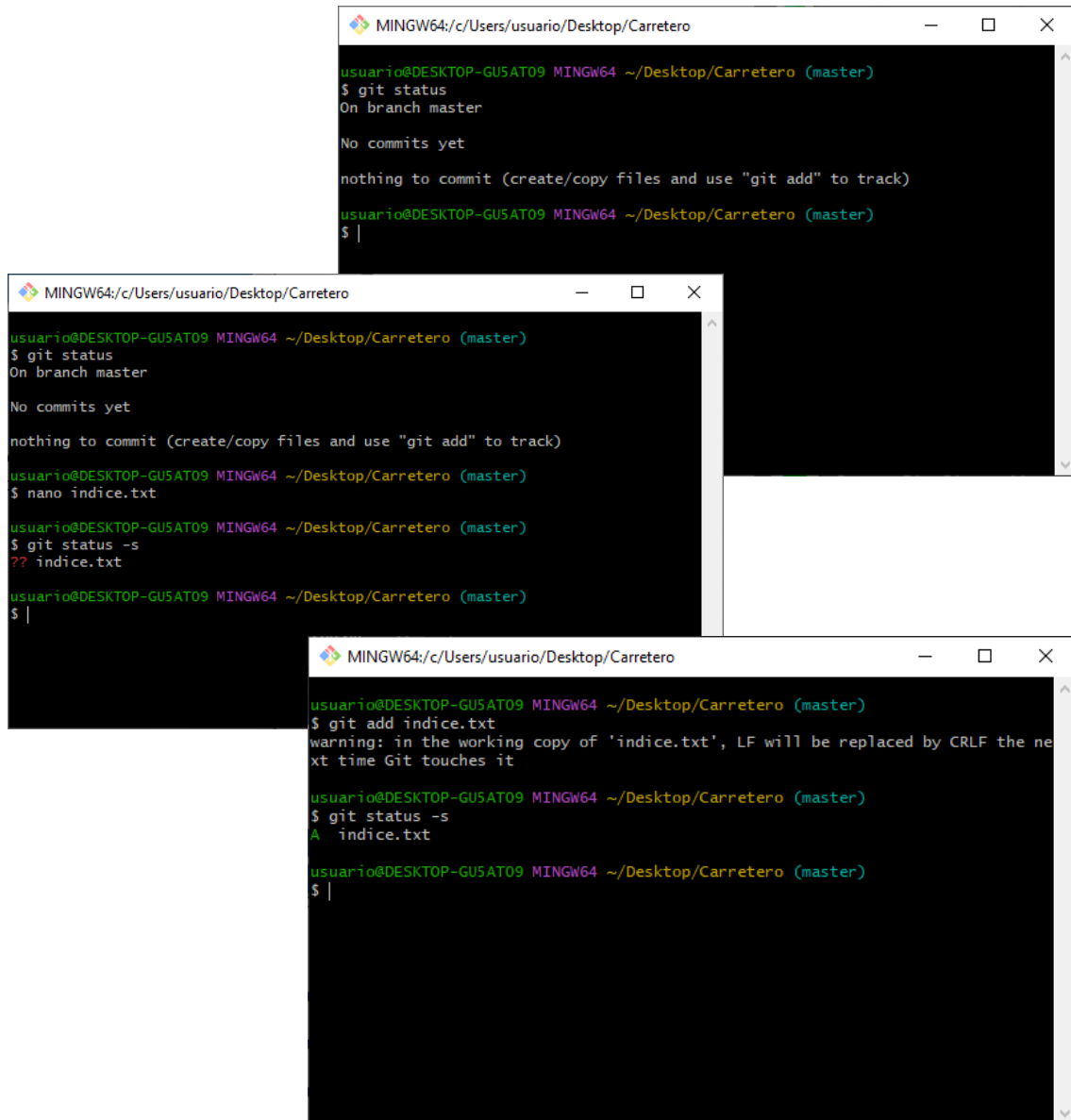
b) Crear un fichero indice.txt con el siguiente contenido:

Capítulo 1: Introducción a Git

Capítulo 2: Flujo de trabajo básico

Capítulo 3: Repositorios remotos

* puedes hacerlo trabajando desde la misma interfaz gráfica del SO o en la misma emulación Bash usando el comando cat o el editor nano.



```
MINGW64:/c/Users/usuario/Desktop/Carretero

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |

MINGW64:/c/Users/usuario/Desktop/Carretero

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ nano indice.txt

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git status -s
?? indice.txt

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |

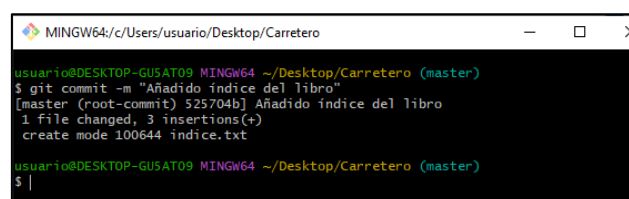
MINGW64:/c/Users/usuario/Desktop/Carretero

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git add indice.txt
warning: in the working copy of 'indice.txt', LF will be replaced by CRLF the next time Git touches it

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git status -s
A  indice.txt

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |
```

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



```
MINGW64:/c/Users/usuario/Desktop/Carretero

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git commit -m "Añadido índice del libro"
[master (root-commit) 525704b] Añadido índice del libro
1 file changed, 3 insertions(+)
create mode 100644 indice.txt

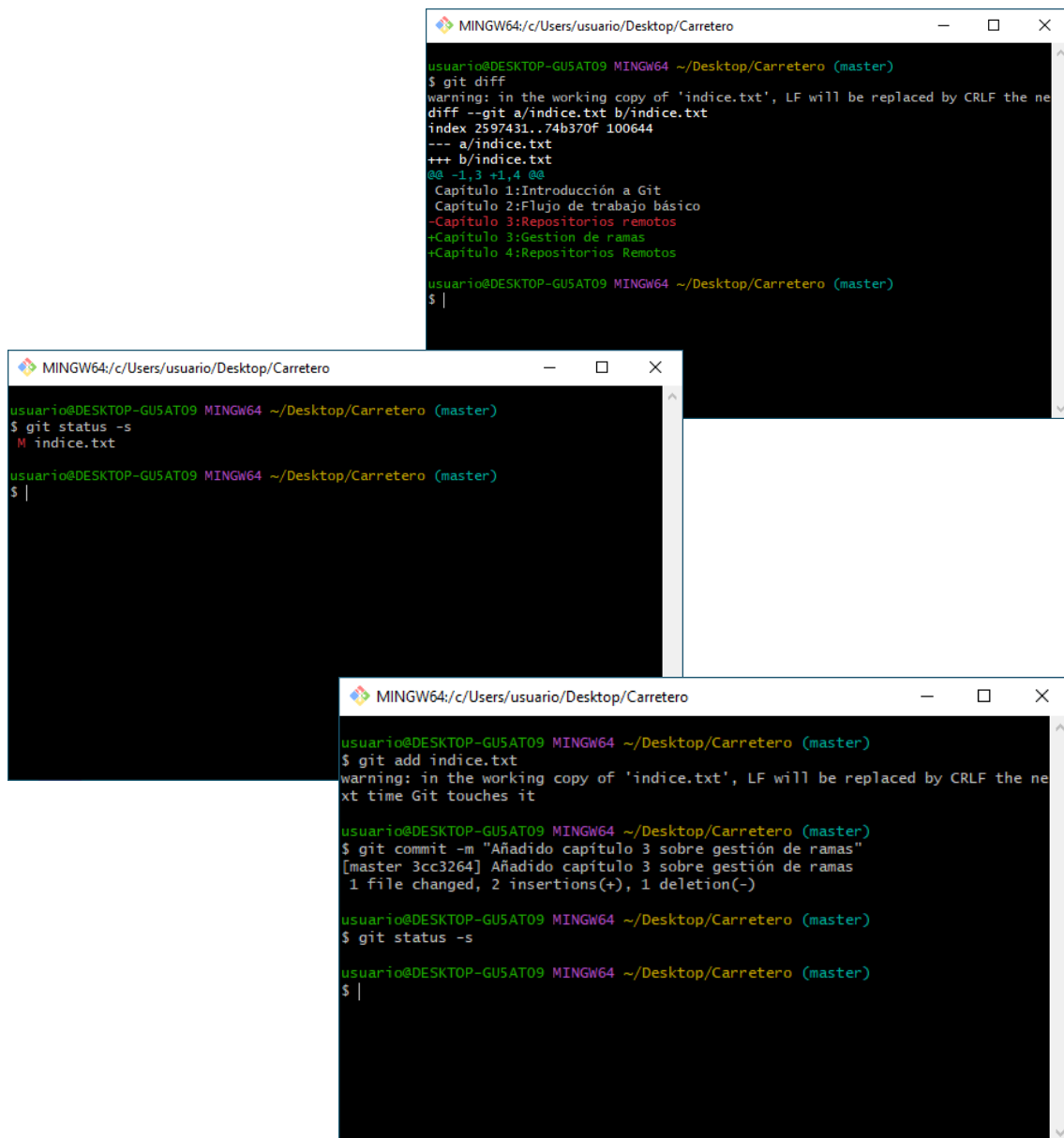
usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |
```

5. 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

* puedes hacerlo trabajando desde la misma interfaz gráfica del SO o en la misma emulación Bash usando el comando cat (en este último caso tendrás que reescribir todo el contenido completo), se aconseja utilizar el editor nano.

- b) Mostrar los cambios con respecto a la última versión guardada en el repositorio.
c) Comprobar el estado del repositorio en formato predeterminado y formato short.
d) Hacer un commit de los cambios con el mensaje “Añadido capítulo 3 sobre gestión de ramas”.
e) Volver a comprobar el estado del repositorio en formato short.



```
MINGW64:/c/Users/usuario/Desktop/Carretero
usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git diff
warning: in the working copy of 'indice.txt', LF will be replaced by CRLF the ne
diff --git a/indice.txt b/indice.txt
index 2597431..74b370f 100644
--- a/indice.txt
+++ b/indice.txt
@@ -1,3 +1,4 @@
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
-Capítulo 3: Repositorios remotos
+Capítulo 3: Gestión de ramas
+Capítulo 4: Repositorios Remotos

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |

MINGW64:/c/Users/usuario/Desktop/Carretero
usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git status -s
M indice.txt

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |

MINGW64:/c/Users/usuario/Desktop/Carretero
usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git add indice.txt
warning: in the working copy of 'indice.txt', LF will be replaced by CRLF the ne
xt time Git touches it

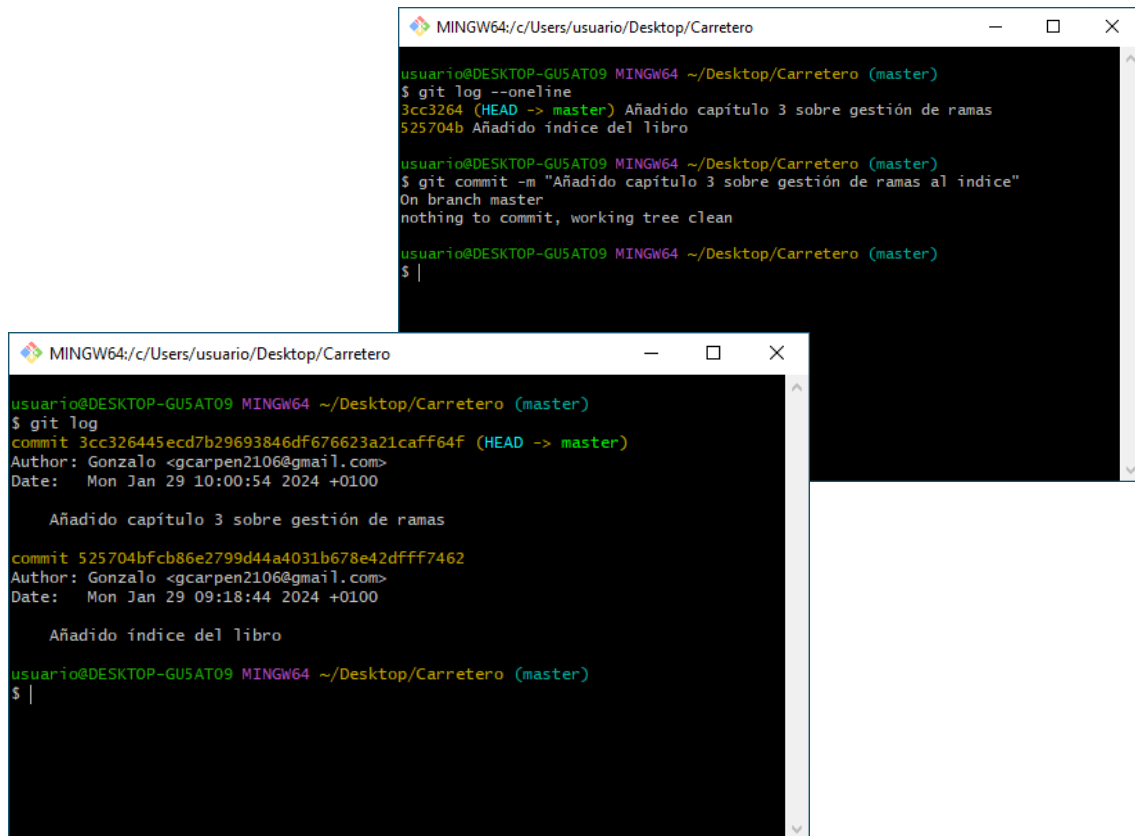
usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git commit -m "Añadido capítulo 3 sobre gestión de ramas"
[master 3cc3264] Añadido capítulo 3 sobre gestión de ramas
1 file changed, 2 insertions(+), 1 deletion(-)

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git status -s

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |
```

6. a) Mostrar los cambios de la última versión del repositorio con respecto a la anterior.
- b) Cambiar el mensaje del último commit por "Añadido capítulo 3 sobre gestión de ramas al índice."
- c) Volver a mostrar los últimos cambios del repositorio.
- d) Mostrar todo el historial de cambios en este repositorio.

Realizar una captura de pantalla y pegarla al fichero imagen de envío final.



```
MINGW64:/c/Users/usuario/Desktop/Carretero

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git log --oneline
3cc3264 (HEAD -> master) Añadido capítulo 3 sobre gestión de ramas
525704b Añadido índice del libro

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git commit -m "Añadido capítulo 3 sobre gestión de ramas al índice"
On branch master
nothing to commit, working tree clean

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |

MINGW64:/c/Users/usuario/Desktop/Carretero

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ git log
commit 3cc326445ecd7b29693846df676623a21caff64f (HEAD -> master)
Author: Gonzalo <gcarpen2106@gmail.com>
Date: Mon Jan 29 10:00:54 2024 +0100

    Añadido capítulo 3 sobre gestión de ramas

commit 525704bfc886e2799d44a4031b678e42dfff7462
Author: Gonzalo <gcarpen2106@gmail.com>
Date: Mon Jan 29 09:18:44 2024 +0100

    Añadido índice del libro

usuario@DESKTOP-GU5AT09 MINGW64 ~/Desktop/Carretero (master)
$ |
```

7. Crear otro nuevo repositorio local con el nombre "capitulos" y mostrar su contenido. Ten en cuenta que cuando abres el emulador Git Bash, de forma predeterminada está trabajando en el home del usuario que instaló la aplicación Git.

a) Crear y guardar un fichero capitulo1.txt con el siguiente texto:

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

* puedes hacerlo trabajando desde la misma interfaz gráfica del SO o en la misma emulación Bash usando el comando cat o el editor nano.

b) Comprobar el estado del repositorio con el formato por defecto y short.

c) Añadir los cambios a la zona staging o de preparación.

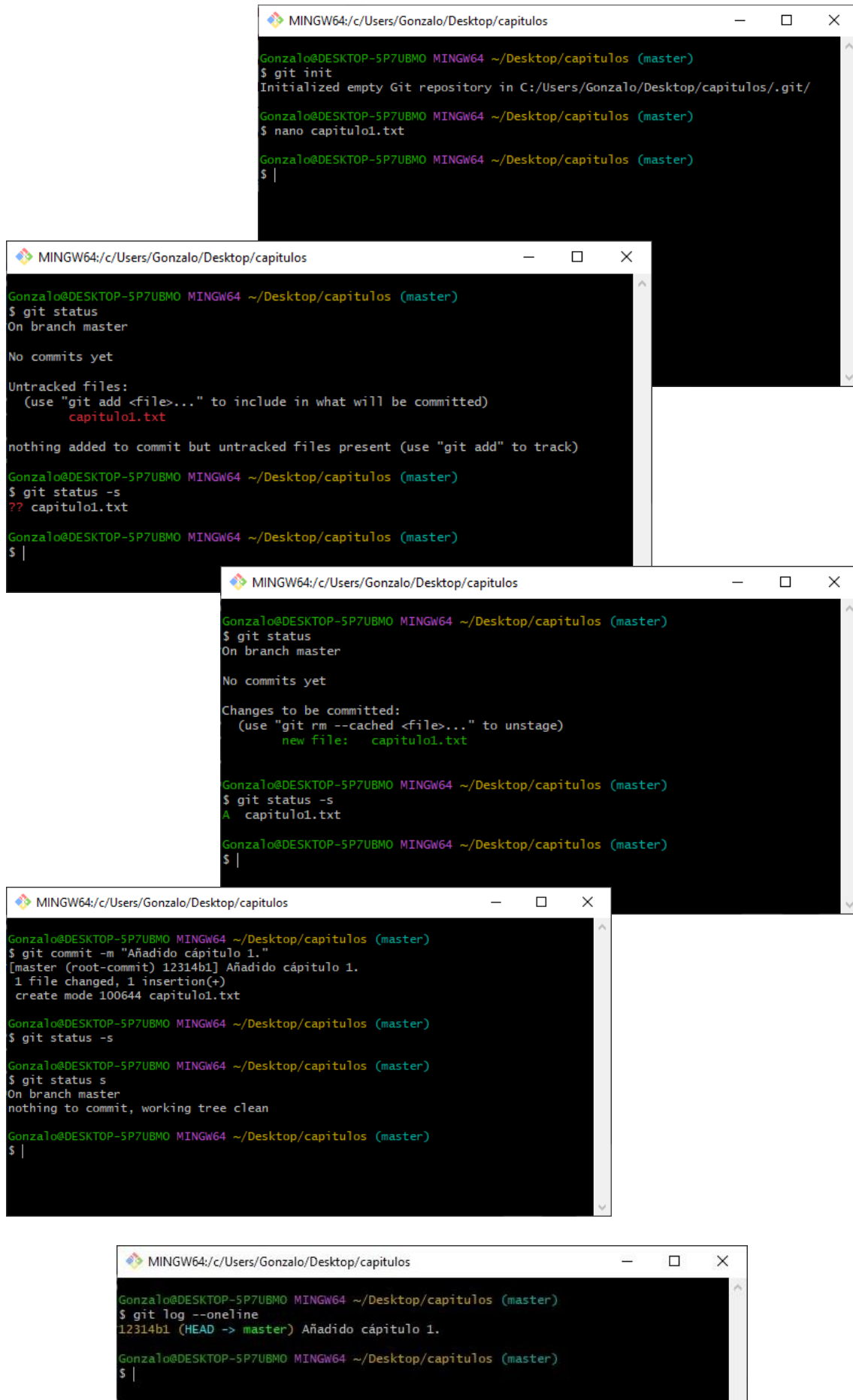
d) Volver a comprobar el estado del repositorio en formato short.

e) Hacer un commit de los cambios con el mensaje "Añadido capítulo 1."

f) Volver a comprobar el estado del repositorio en formato predeterminado.

g) Mostrar el historial de cambios del repositorio.

Realizar una captura de pantalla y pegarla al fichero imagen de envío final.



```
MINGW64:/c/Users/Gonzalo/Desktop/capitulos

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git init
Initialized empty Git repository in C:/Users/Gonzalo/Desktop/capitulos/.git/

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ nano capitulo1.txt

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ |

MINGW64:/c/Users/Gonzalo/Desktop/capitulos

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        capitulo1.txt

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

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git status -s
?? capitulo1.txt

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ |

MINGW64:/c/Users/Gonzalo/Desktop/capitulos

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   capitulo1.txt

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git status -s
A  capitulo1.txt

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ |

MINGW64:/c/Users/Gonzalo/Desktop/capitulos

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git commit -m "Añadido capítulo 1."
[master (root-commit) 12314b1] Añadido capítulo 1.
1 file changed, 1 insertion(+)
create mode 100644 capitulo1.txt

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git status -s

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git status s
On branch master
nothing to commit, working tree clean

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ |

MINGW64:/c/Users/Gonzalo/Desktop/capitulos

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git log --oneline
12314b1 (HEAD -> master) Añadido capítulo 1.

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ |
```

8. a) 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.
- * puedes hacerlo trabajando desde la misma interfaz gráfica del SO o en la misma emulación Bash usando el comando cat.
- b) Añadir los cambios a la zona de preparación.
- c) Hacer un commit de los cambios con el mensaje "Añadido capítulo 2."
- d) Mostrar el historial del repositorio resumida en una sola línea por cada versión (--oneline)

```
MINGW64:/c/Users/Gonzalo/Desktop/capitulos
GNU nano 7.2 capitulo2.txt Modified
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

MINGW64:/c/Users/Gonzalo/Desktop/capitulos
Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git add capitulo2.txt
warning: in the working copy of 'capitulo2.txt', LF will be replaced by CRLF the next time Git touches it

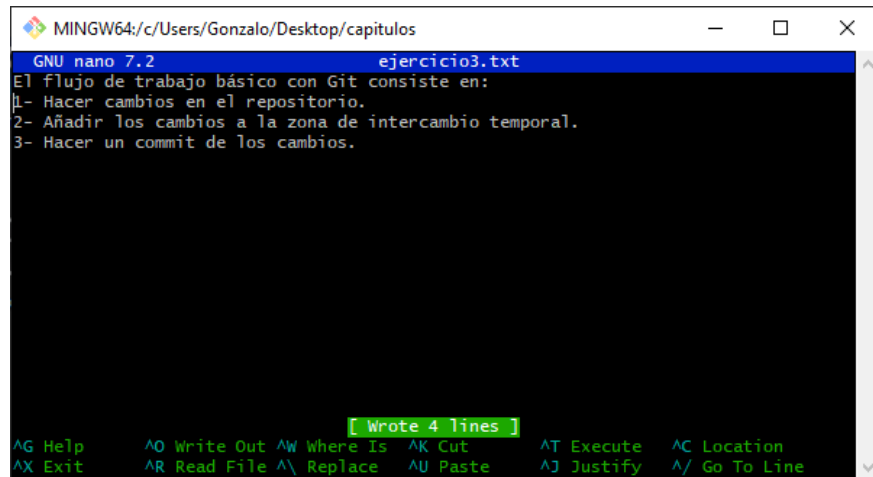
Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git commit -m "Añadido capitulo2"
[master ff3ea9e] Añadido capitulo2
1 file changed, 4 insertions(+)
create mode 100644 capitulo2.txt

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$

MINGW64:/c/Users/Gonzalo/Desktop/capitulos
Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git log --oneline
ff3ea9e (HEAD -> master) Añadido capitulo2
12314b1 Añadido capítulo 1.

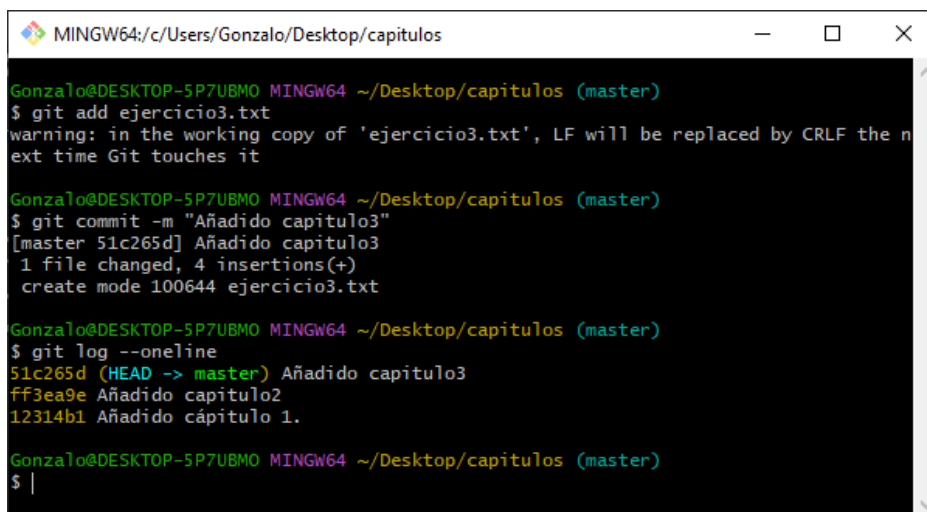
Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$
```

9. Repite el ejercicio 8 creando y guardando un nuevo fichero `capitulo3.txt` con la siguiente información:
- Git permite la creación de ramas lo que permite tener distintas versiones del mismo [proyecto](#) y trabajar de manera simultanea en ellas.
- Una vez hayas guardado todas instantáneas en el repositorio, mostrar el historial del repositorio resumida en una sola línea por cada versión (`--oneline`) y las diferencias entre la última versión y dos versiones anteriores.



```
MINGW64:/c/Users/Gonzalo/Desktop/capitulos
GNU nano 7.2 ejercicio3.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.

[Wrote 4 lines]
```



```
MINGW64:/c/Users/Gonzalo/Desktop/capitulos

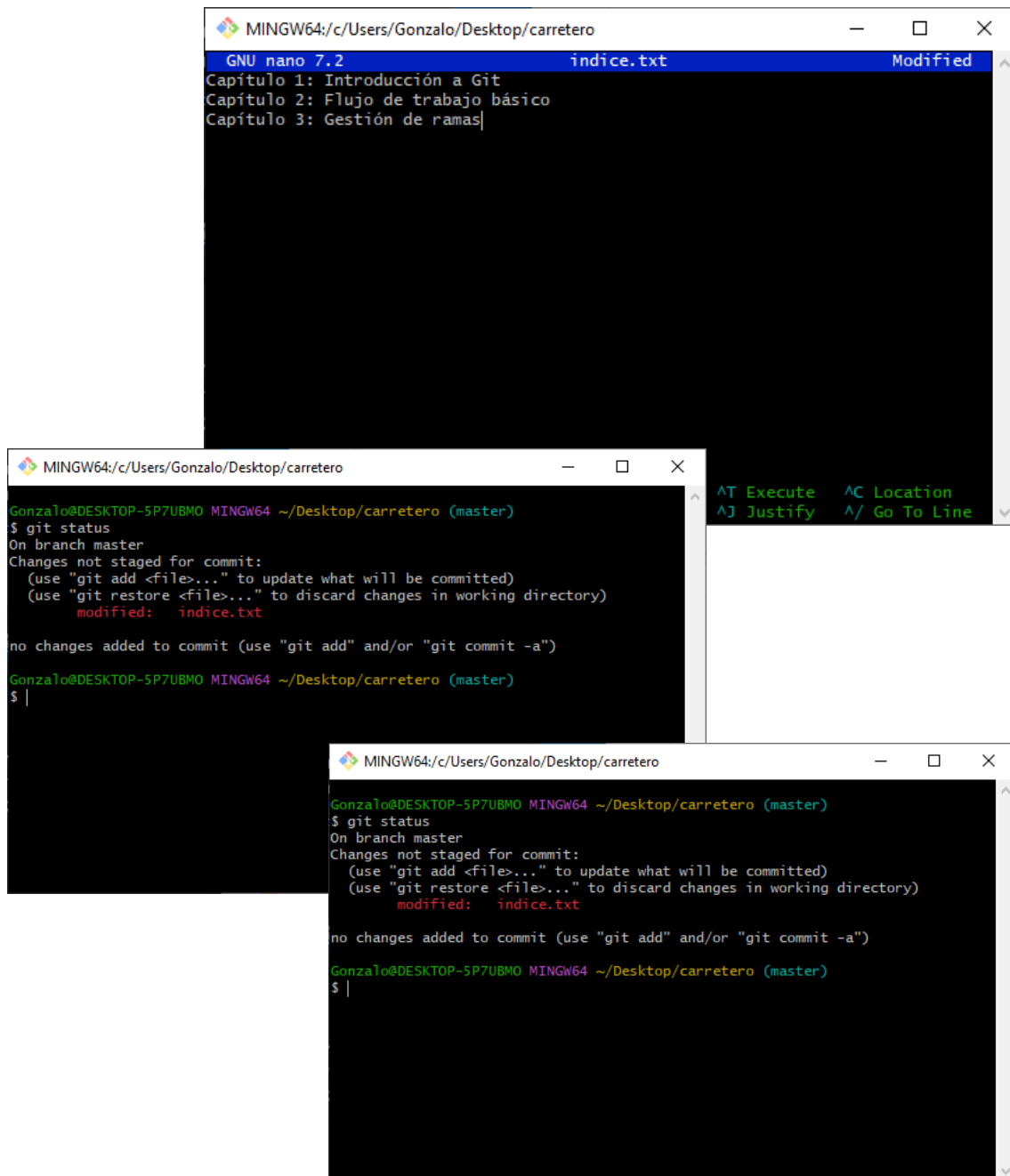
Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git add ejercicio3.txt
warning: in the working copy of 'ejercicio3.txt', LF will be replaced by CRLF the next time Git touches it

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git commit -m "Añadido capitulo3"
[master 51c265d] Añadido capitulo3
1 file changed, 4 insertions(+)
create mode 100644 ejercicio3.txt

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ git log --oneline
51c265d (HEAD -> master) Añadido capitulo3
ff3ea9e Añadido capitulo2
12314b1 Añadido capítulo 1.

Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/capitulos (master)
$ |
```

10. Volver al repositorio primer apellido del ejercicio 2 .
- Eliminar la última línea del fichero `indice.txt` y guardarlo.
* puedes hacerlo trabajando desde la misma interfaz gráfica del SO o en la misma emulación Bash usando el editor nano.
 - Comprobar el estado del repositorio.
 - Deshacer los cambios realizados en el fichero `indice.txt` para volver a la versión anterior del fichero.
 - Volver a comprobar el estado del repositorio.



The image shows three overlapping screenshots of a Windows terminal window. The top window shows a nano editor editing 'indice.txt' with the following content:

```
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
```

The middle window shows the output of the 'git status' command:

```
Gonzalo@DESKTOP-5P7UBMO MINGW64 ~/Desktop/carretero (master)
$ git status
On branch 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:   indice.txt

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

The bottom window shows the same 'git status' command output, but with a different cursor position at the end of the line.

EJERCICIO 11

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



11. Volvemos de nuevo al repositorio capitulos. Después de analizar nuestro proyecto nos damos cuenta que no nos gusta los capítulos 2 y 3, queremos volver a la versión del proyecto desde el capítulo 1.
 Importante: una vez el puntero apunto a la versión grabada del capítulo 1, perderemos todos las demás versiones.
 - a) Mostrar primero el historial del repositorio resumida en una sola línea por cada versión (`--oneline`). Tomar nota de todos los códigos hash porque se necesitarán para seguir con la actividad.

- b) Comprobar el contenido del proyecto con la primera versión grabada del capítulo trasladando el puntero HEAD a su código hash correspondiente.
- c) Trasladar de nuevo el puntero head a su último commit, es decir, insertado capítulo 3.
- e) Volver a la versión inicial del proyecto pero esta vez borrando las dos versiones finales 2 y 3.
- f) Mostrar una vez más el historial del repositorio resumida en una sola línea por cada versión (--oneline).