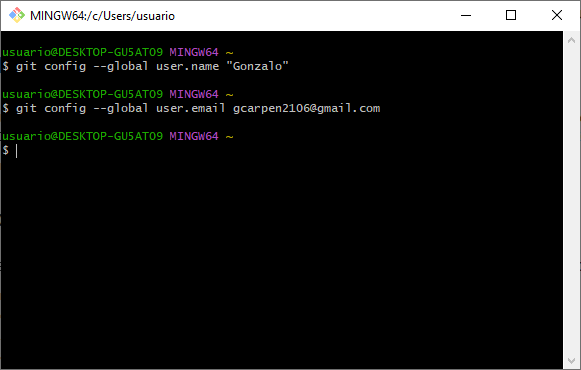
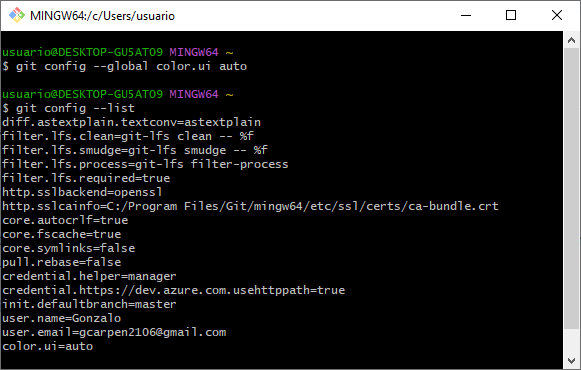
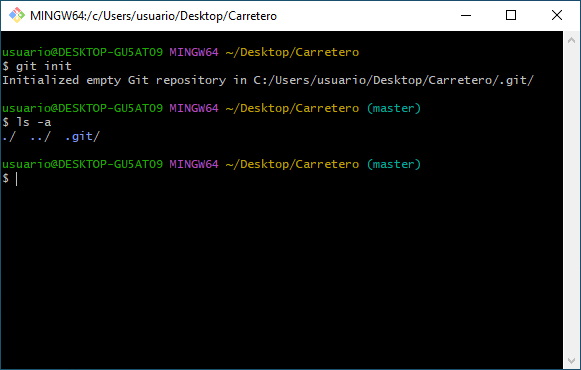
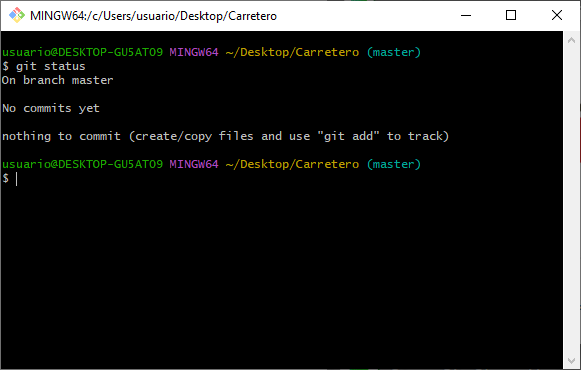
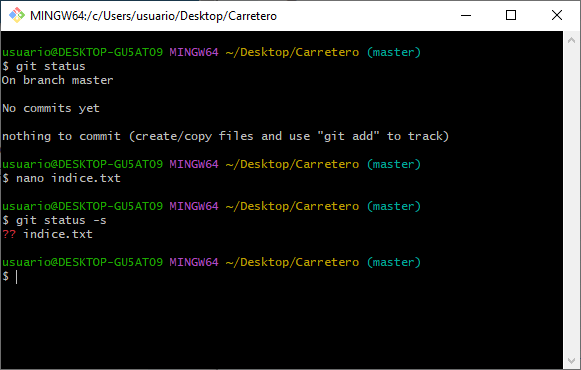
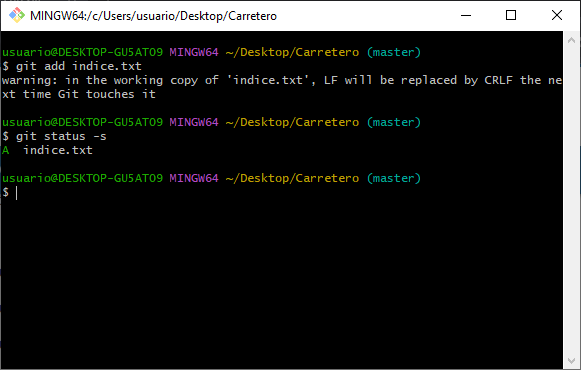
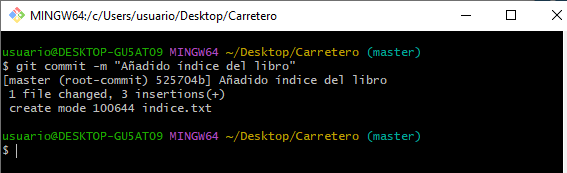
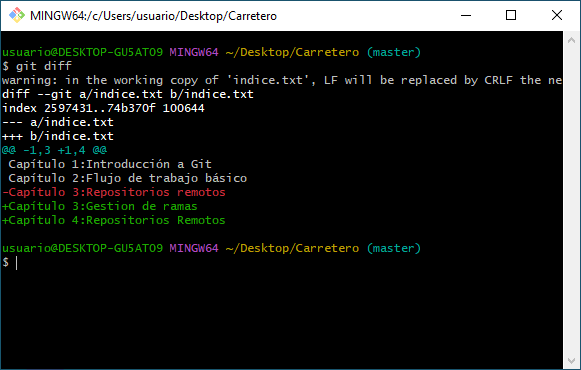
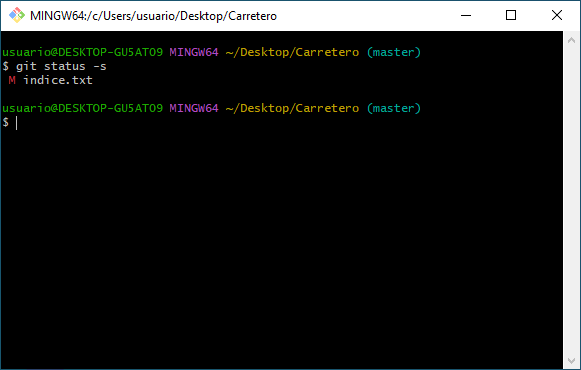
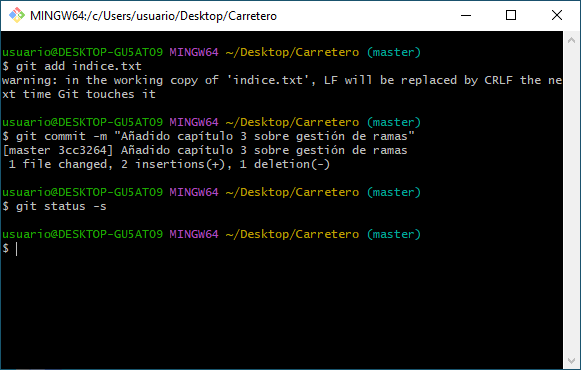
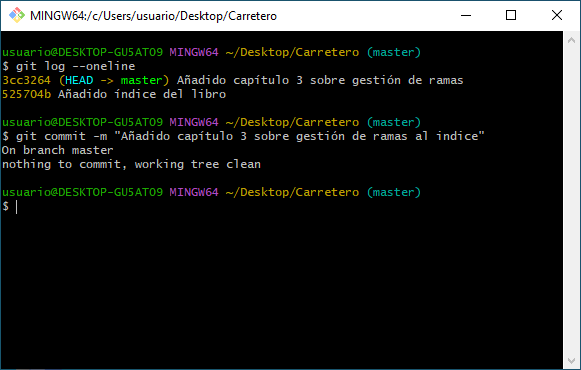
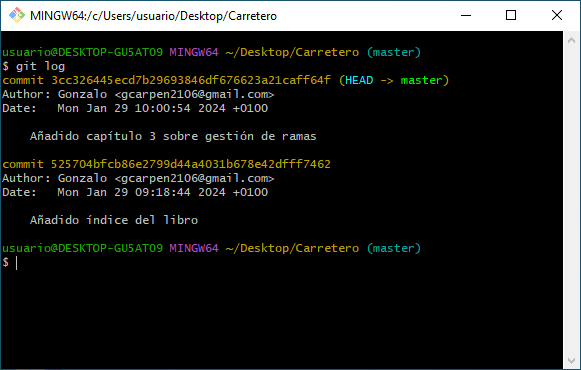
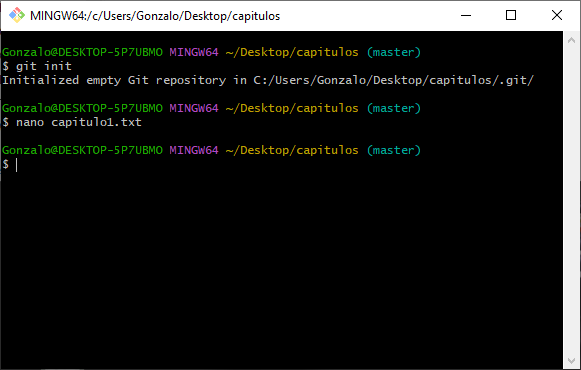
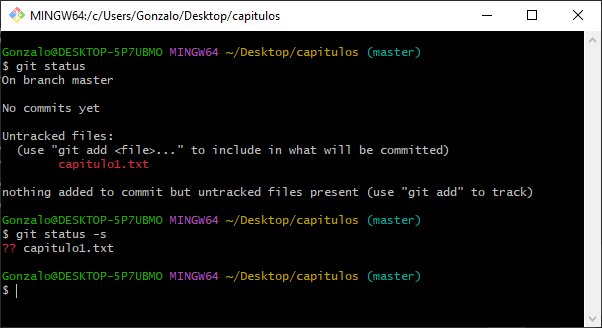
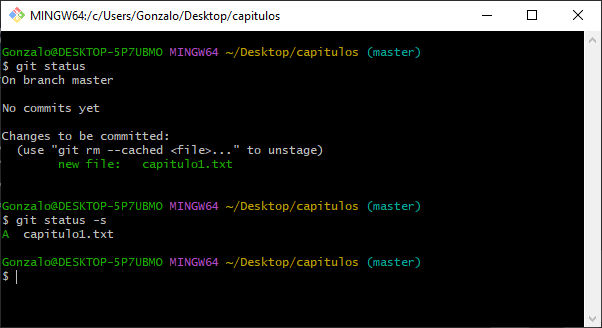
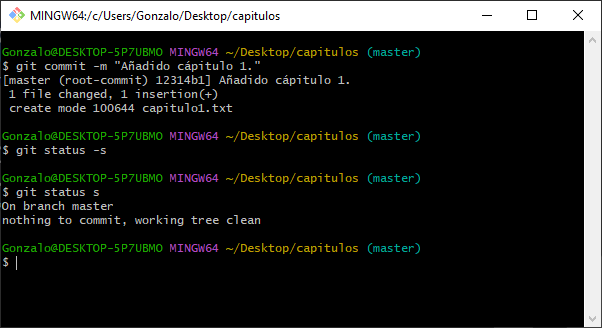
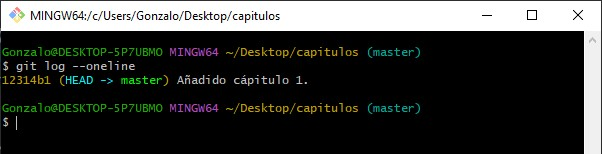
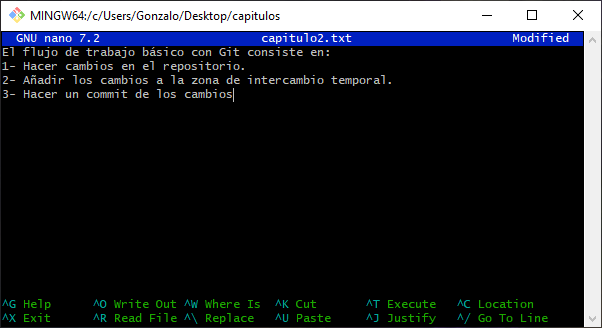
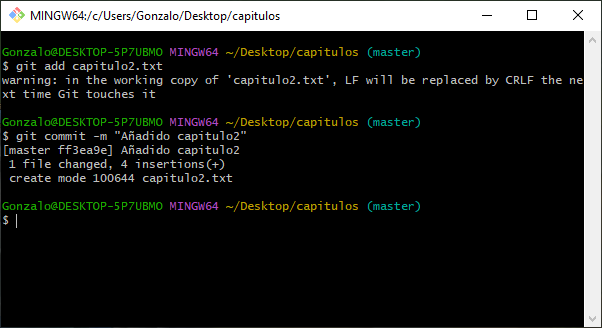
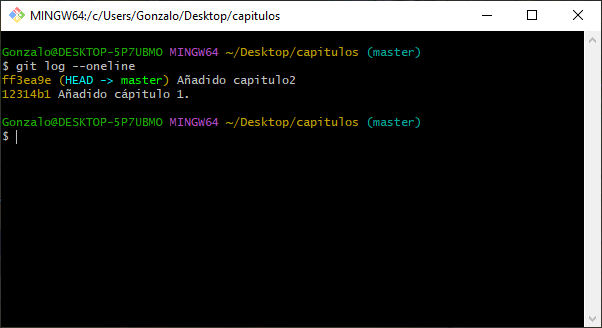
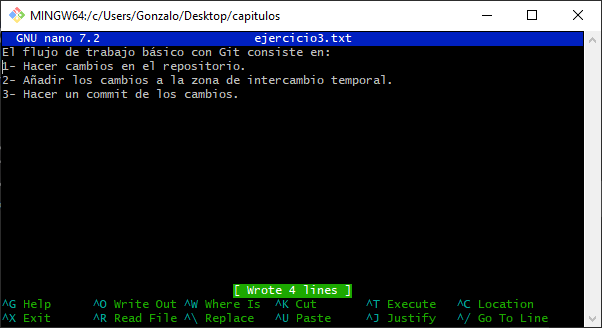
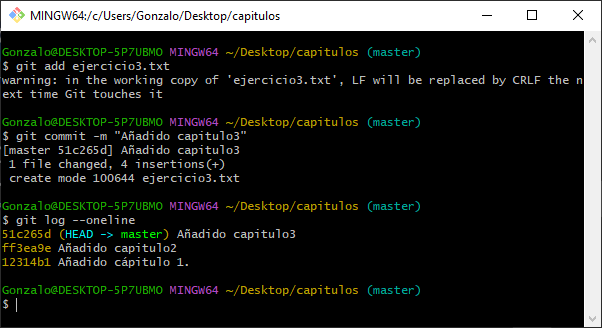
1. Instalar Git en vuestro sistema. Una vez instalado, si utilizáis Windows utilizar el emulador Git Bash para configurarlo con vuestro nombre de usuario, vuestro correo electrónico g.educaand.es. Git puede marcar con colores los resultados que muestra en tu terminal, ayudándote así a leerlos más fácilmente.,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



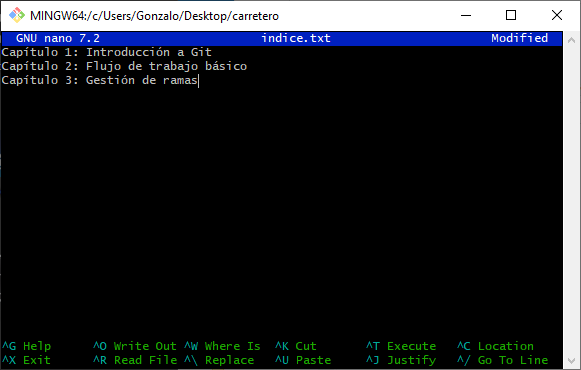
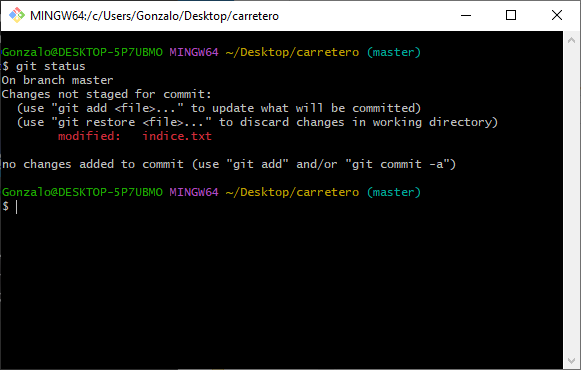
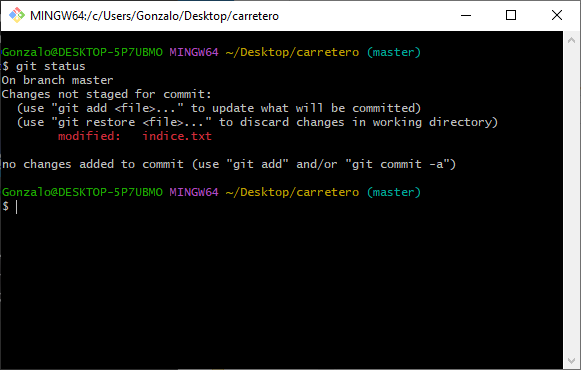
1. **Cambiar el fichero**indice.txt**para que contenga lo siguiente:**  
   **Capítulo 1: Introducción a Git**  
   **Capítulo 2: Flujo de trabajo básico**  
   **Capítulo 3: Gestión de ramas**  
   **Capítulo 4: Repositorios remotos**  
   **\* puedes hacerlo trabajando desde la misma interfaz gráfica 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.
2. 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.
3. 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.



1. 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 )
2. 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](https://educacionadistancia.juntadeandalucia.es/centros/cadiz/mod/assign/view.php?id=383569) 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.

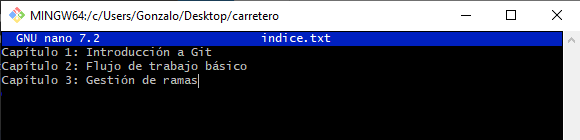
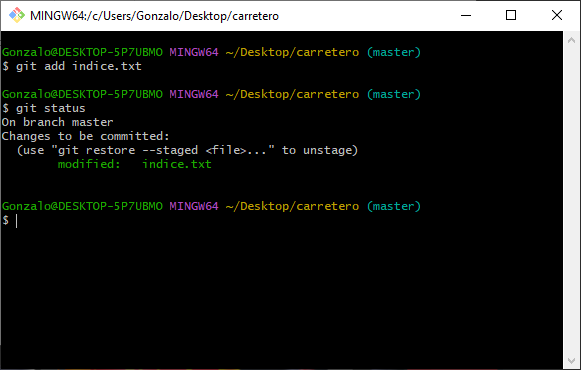
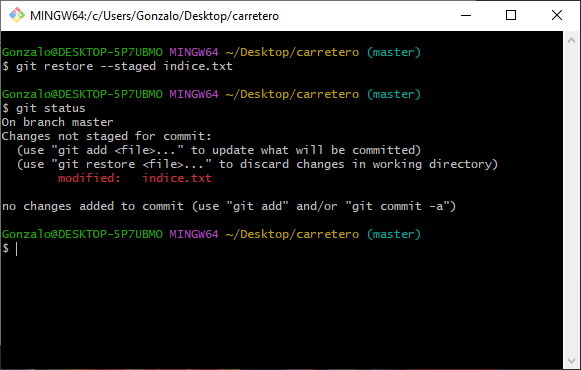
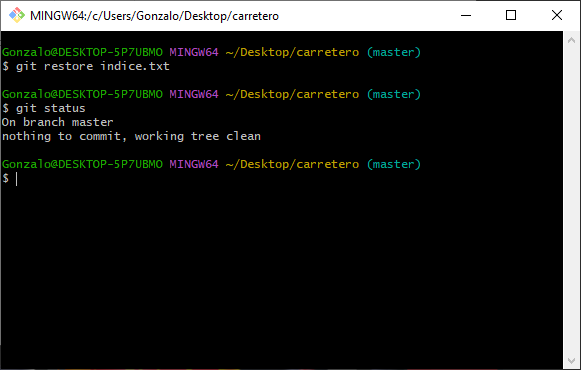


1. Volver al repositorio primer apellido del ejercicio 2 .  
   a) Eliminar la última línea del fichero indice.txt y guardarlo.  
   \* puedes hacerlo trabajando desde la misma interfaz gráfica del SO o en la misma emulación Bash usando el editor nano.  
   b) Comprobar el estado del repositorio.  
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



**EJERCICIO 11**

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



1. 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 ).