

Apartado 1: Conociendo los comandos básicos de Git

1º) Tras visualizar el vídeo, contesta a las siguientes preguntas:

a) ¿Para qué sirven estos comandos de Git?

- **git init:** *se utiliza cuando comenzamos el proyecto y solo se ejecuta una vez. Sirve para indicarle a Git que haga un seguimiento de todos nuestros archivos o de parte de ellos. Además, crea dos áreas donde irá almacenando los archivos: “Área de ensayo” y “Repositorio local”.*
- **git add:** *se utiliza para indicarle a Git los archivos concretos a los que queremos que les realiza el seguimiento. Ejecutándolo, conseguimos que Git lleve ese archivo desde nuestro directorio de trabajo al área de ensayo.*
- **git commit:** *traslada el archivo o los archivos que tengamos en el área de ensayo al área del repositorio local donde Git toma la “instantánea” del archivo. Una vez tenemos en el repositorio local el archivo o archivos, podemos obtener un respaldo de cómo se encontraba el repositorio local en cualquier momento.*
- **git commit -m:** *funciona igual que “git commi” pero añade una descripción de los archivos al guardado que realiza Git.*
- **git status:** *nos proporciona un listado con todos los archivos que enemos de momento en la carpeta del proyecto.*

b) ¿Dónde puedo consultar la función de todos los comandos de Git?

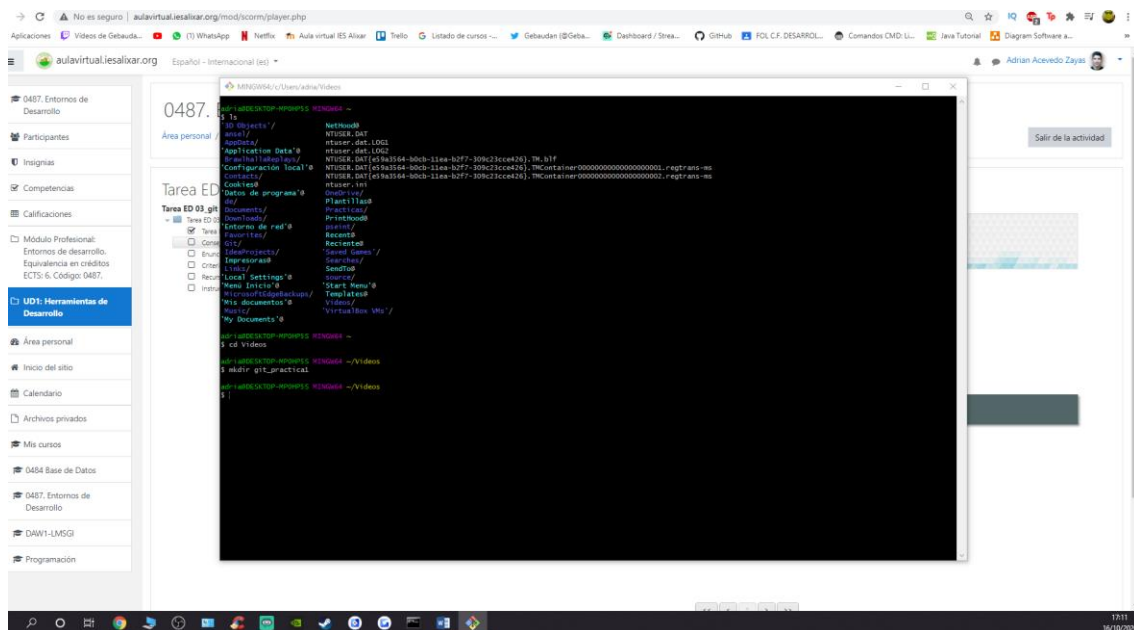
Podemos encontrar esa información en el apartado “Documentation” de la página web de Git. Dentro de este apartado, haremos click sobre “Reference Manual” y a continuación, haremos click de nuevo sobre “Complete list of all commands”.

Apartado 2: Poniendo en práctica los comandos básicos

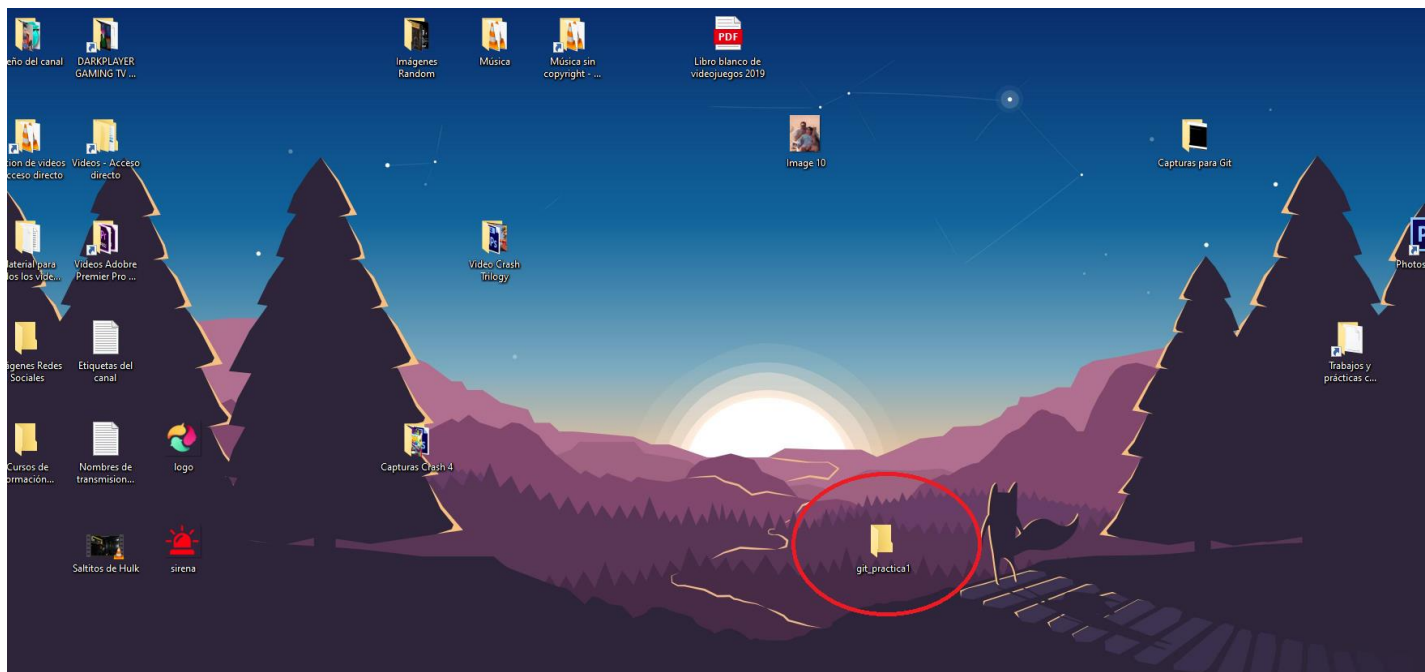
2º) Por cada pregunta, debes:

- Indicar el comando o los comandos que has usado en cada pregunta.
- Aportar al menos una captura (siguiendo las instrucciones de entrega).
- Razonar y explicar la información que te devuelve la consola.

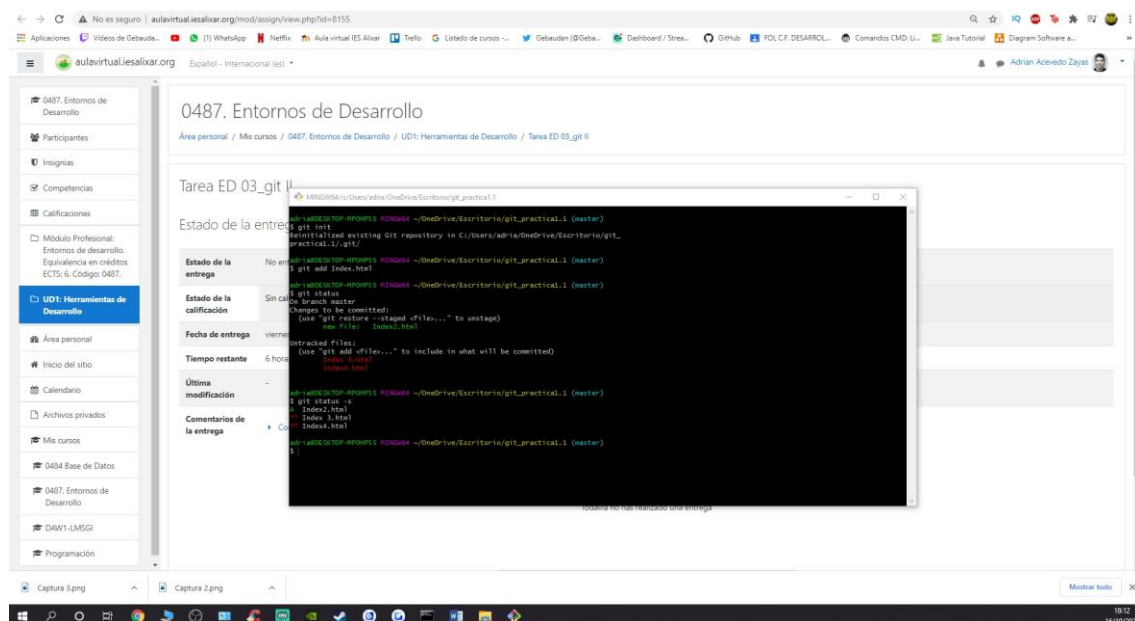
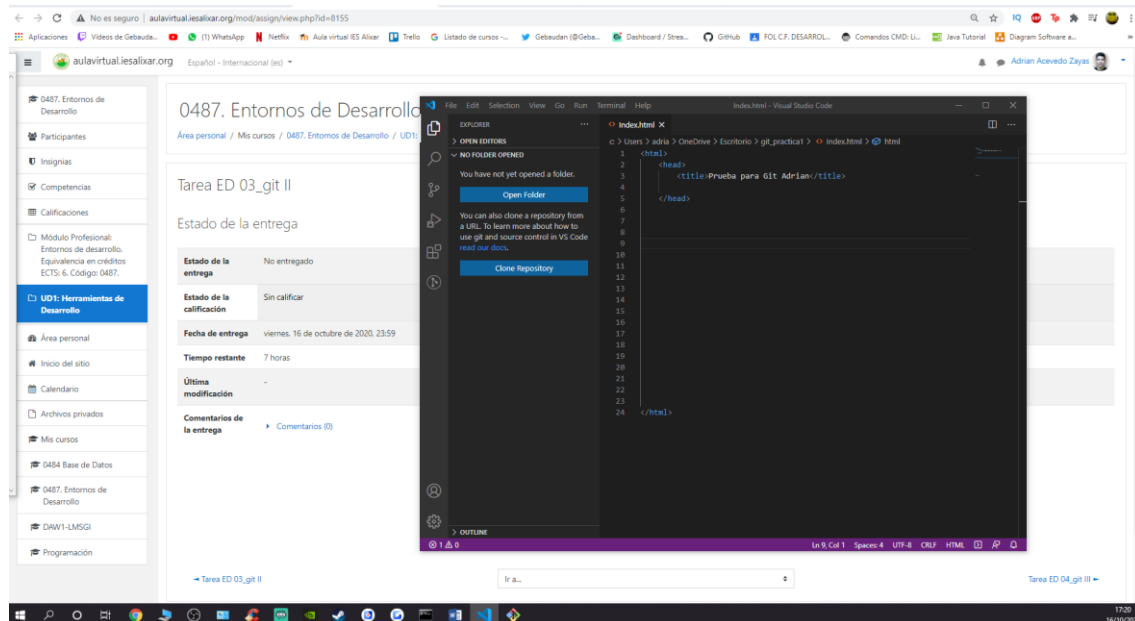
1) Crea un directorio en tu ordenador “git_practica1”.



He utilizado el comando “mkdir” para crear la carpeta. En un principio la creé en mi carpeta “Videos” pero luego, reinicié la práctica y la creé en mi escritorio:

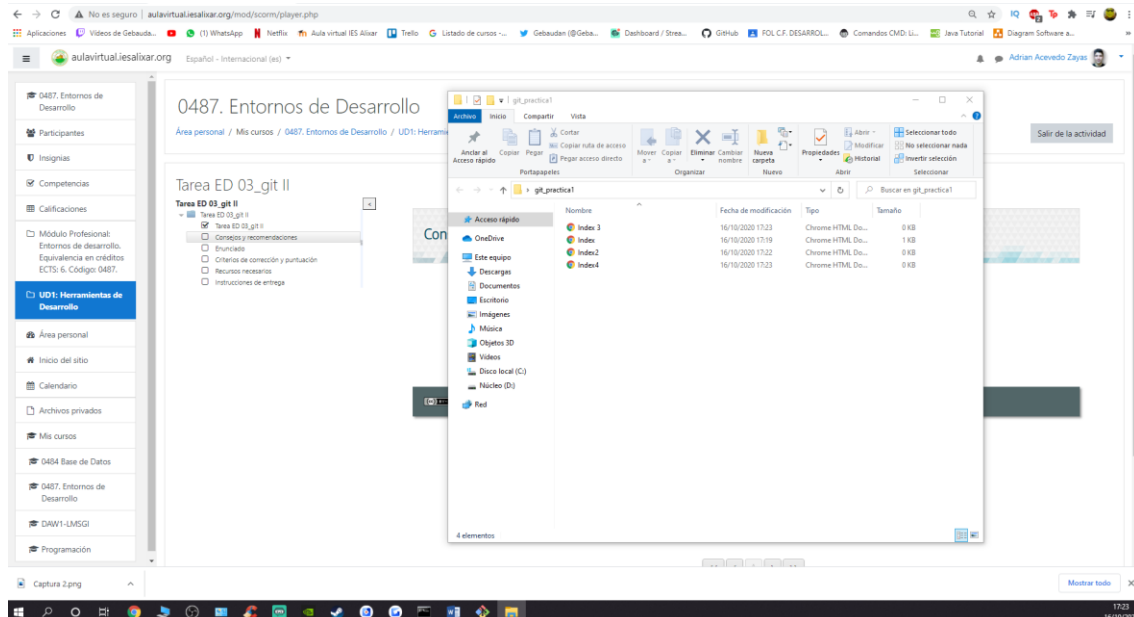


2) Añade un archivo “index.html” y escribe en él algún código (por ejemplo, un encabezado)

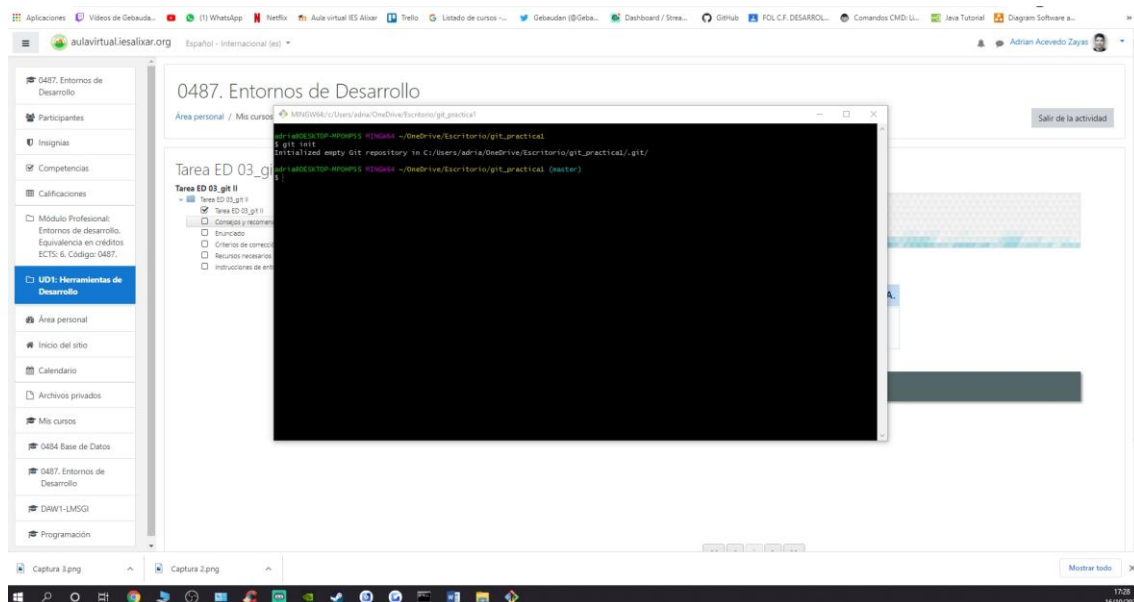


Para añadir el archivo he usado el comando “git add” + el nombre del archivo. En la captura se aprecia que he añadido el archivo “Index 2.html” no el 1 ya que, revisando el trabajo, no hice captura de ese paso y lo volví a hacer con el archivo 2 para mostrarlo.

3) Crea varios directorios más dentro de “git_practica1” y guarda en ellos varios archivos html más (no hace falta que tengan código).

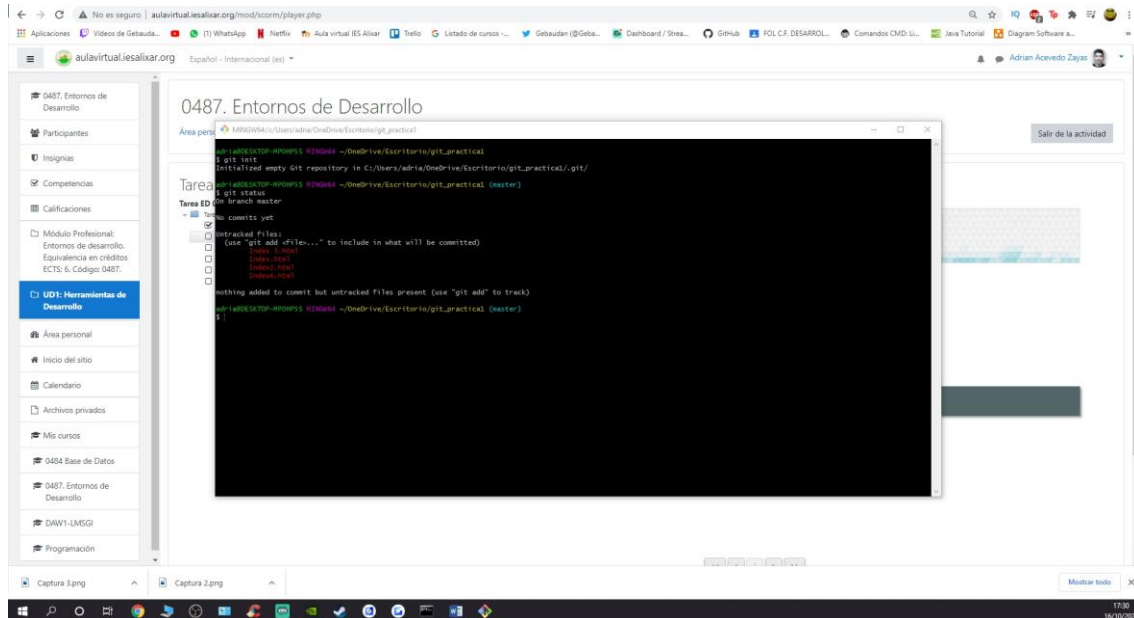


4) Inicia git en este directorio.



Utilizo el comando “git init” para comenzar el proyecto. Tras hacerlo, la máquina me indica: “Iniciado repositorio de Git vacío en (localización)”

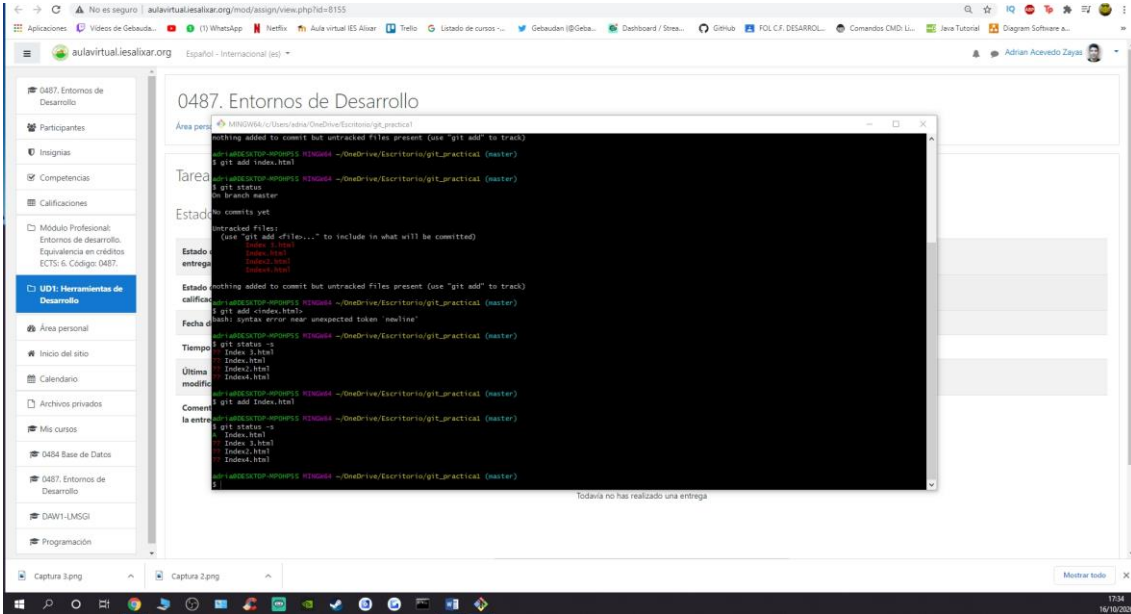
5) Comprueba el estado de todos los archivos y directorios del proyecto git. Explica en qué área de git está cada archivo.



He utilizado el comando “git status” para comprobar el estado del proyecto. Como bien nos indica la consola, aún no se ha hecho un commit. Habría que utilizar el comando “git add <archivo>” para incluirlo en el commit. También indica que no se ha añadido nada al commit pero se han encontrado archivos sin seguimiento o copia.

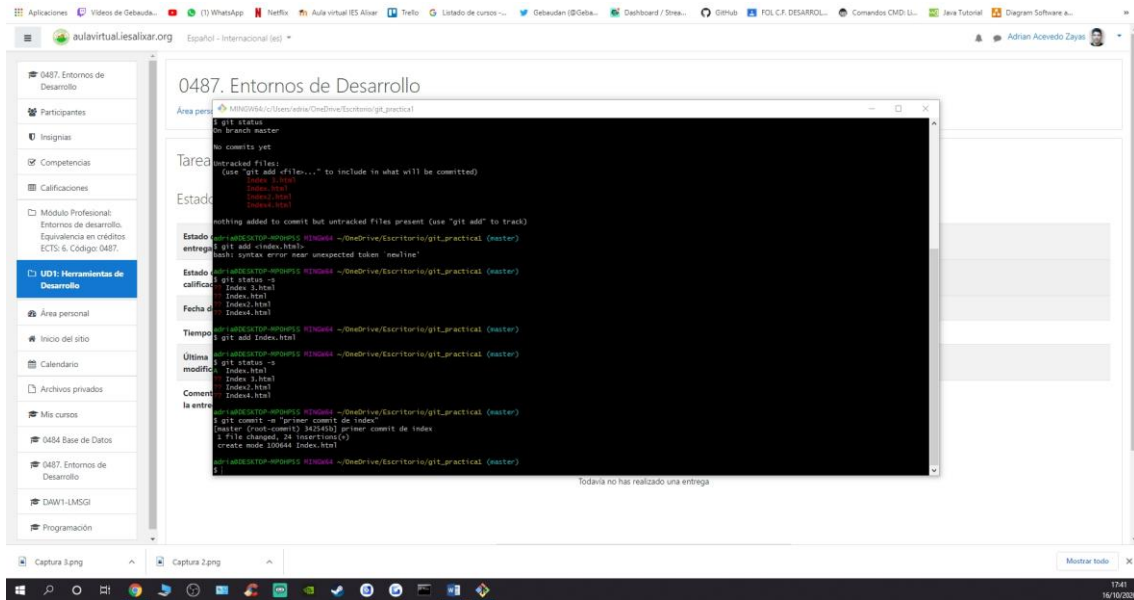
Dado que hicimos “Git init” esos archivos se encuentran en el Área de ensayo, a la espera de ser añadidos para subirlos con el commit.

6) Añade el archivo index a git y vuelve a comprobar el estado de los archivos, ¿qué diferencia hay con la anterior comprobación? Explícalo.



He usado el comando “git add” para añadir el archivo “Index.html”. Hecho esto, el archivo añadido aparece en verde, indicando que ya está añadido al commit para realizar la copia una vez se lo indiquemos. Añado también el comando “git status”, ligeramente distinto a “git status -s” como indica el video.

7) Confirma a git que el código del archivo index está ok (de momento). Para ello, indica esto en el mensaje (por ejemplo “primer commit de index”)

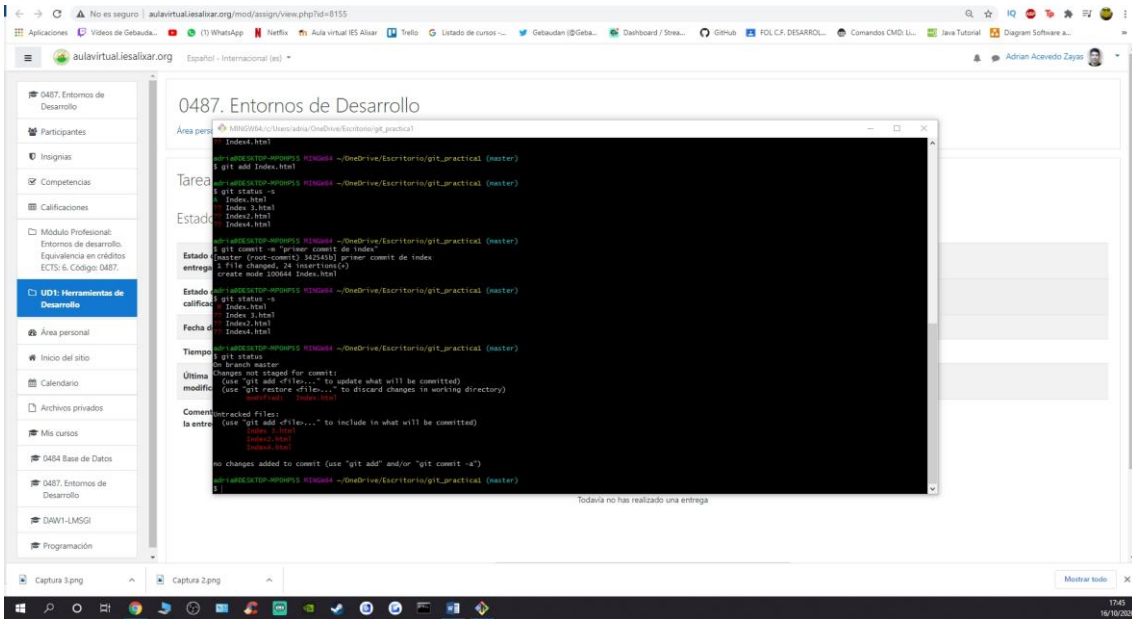


He utilizado el comando “git commit –m” seguido del mensaje indicado en el enunciado para hacerle la primera copia de versión al archivo. Indicar también que “git commit” se diferencia de “git commit –m” en que en este último va con una indicación mientras que el anterior realiza la acción directamente.

La consola en esta ocasión nos indica lo siguiente:

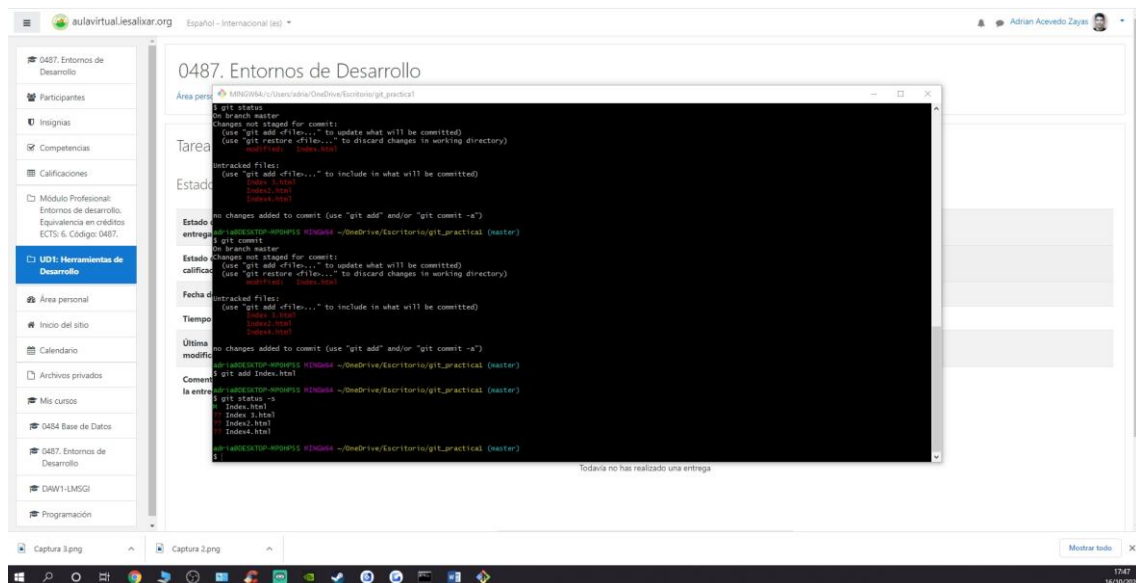
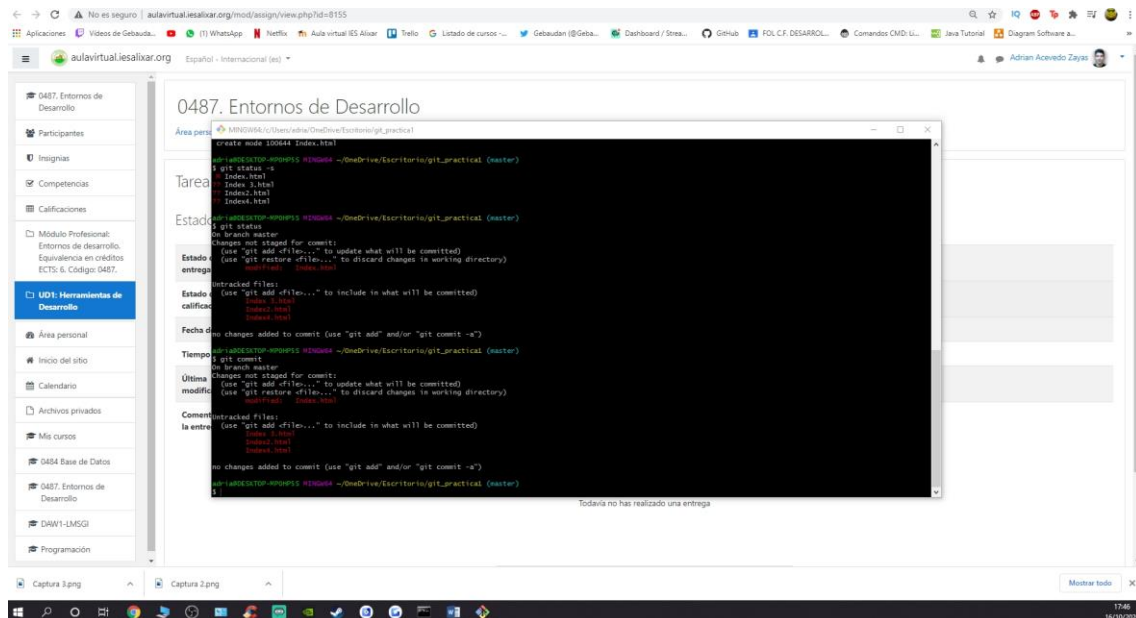
- *“Un archivo se ha modificado, creado el mod 100644 para Index.html”*

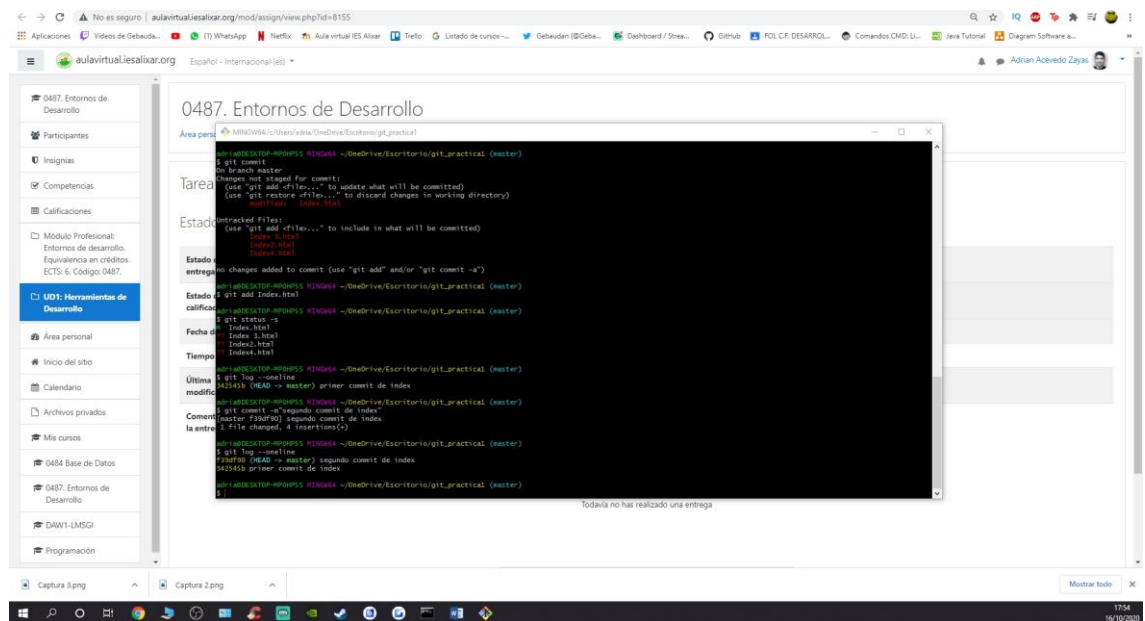
8) **Añade más código a tu archivo index, por ejemplo, varios párrafos con texto y vuelve a comprobar el estado de los archivos, ¿qué diferencia hay con la anterior comprobación? Explícalo.**



Nuevamente usamos el comando “git status” para comprobar el estado de los archivos. En esta ocasión, como hemos modificado el archivo Index.html, git nos indica en rojo que, efectivamente, el archivo index.html ha sido modificado y que, para tener la última versión del mismo, debemos añadirlo con el comando “git add” para subirlo posteriormente con un “git commit”.

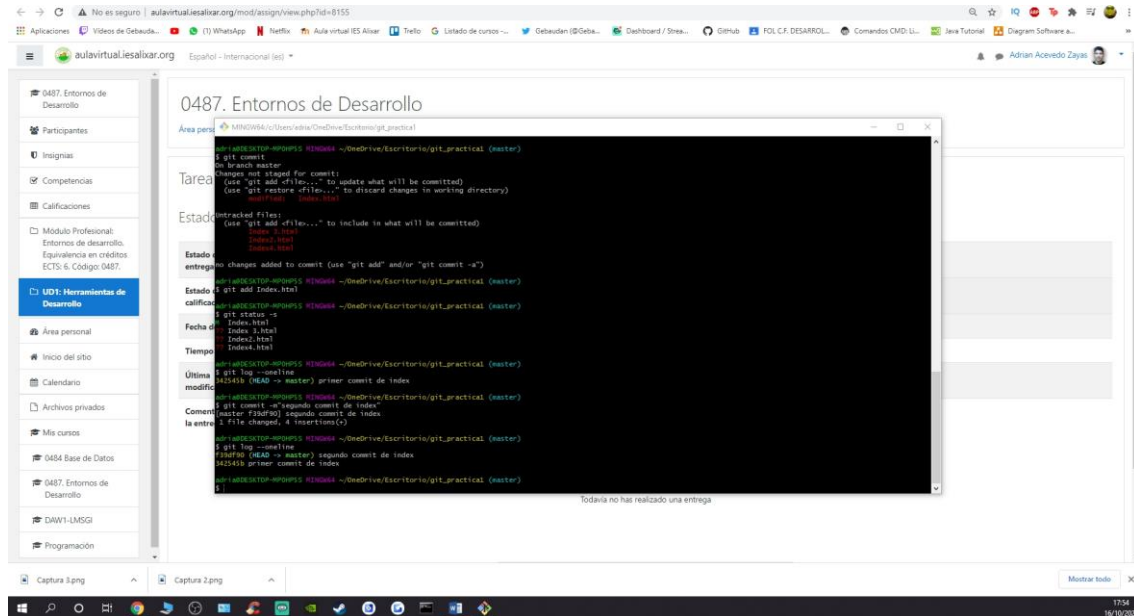
9) Intenta hacer un commit (no creo que puedas), explica qué pasa y soluciona el error para poder hacerlo.





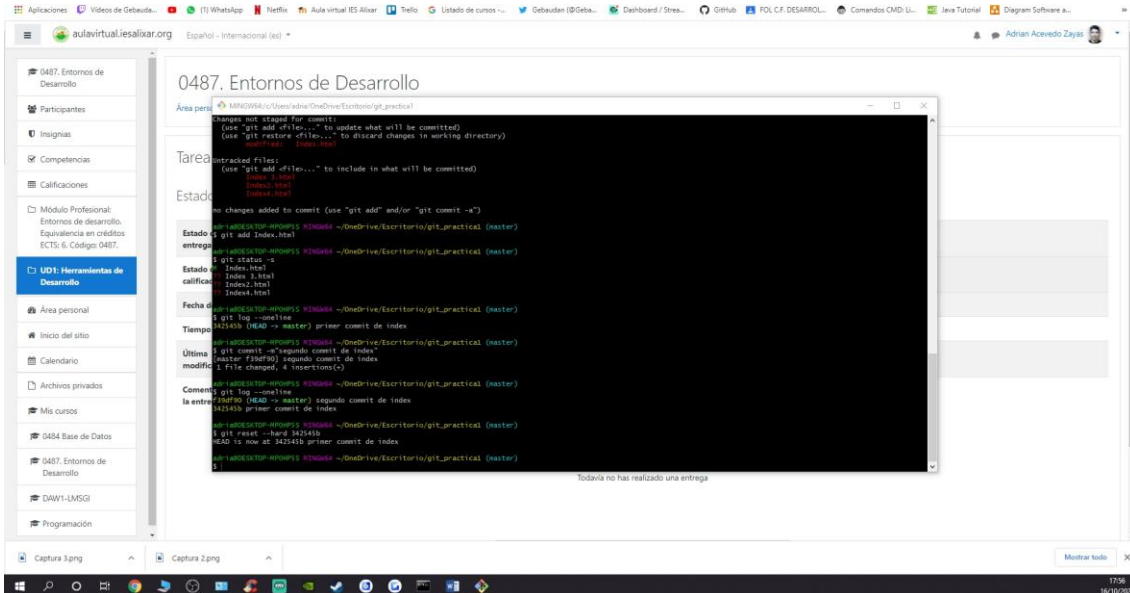
Adjunto 3 capturas del proceso secuencial; utilizo el comando “git status” de nuevo para conocer el estado de los archivos. Como no habríamos añadido el archivo Index.html sin los cambios, git nos indica que no hay cambios añadidos al commit por lo que la versión de nuestro proyecto no cambia. Para solucionarlo, debemos añadir de nuevo el archivo ya modificado con “git add” y, luego si, realizar el “git commit” con la modificación ya realizada.

10) Comprueba cuántas copias de index tienes en tu repositorio.



Para comprobar cuantas copias tenemos de nuestro archivo, utilizaremos el comando “git log –oneline”. En la captura, lo ejecuto una vez para comprobar que solo tengo realizado un commit. Tras verificarlo, realizo el segundo commit con el archivo ya modificado. A continuación, ejecuto de nuevo el comando “git log –oneline” para verificar que tendría 2 copias de Index en mi repositorio.

11) Restaura tu proyecto con el primer index que codificaste.



Para restaurar el proyecto con la primera versión de nuestro archivo `Index.html`, utilizaremos el comando `"git reset --hard"`. Esto nos llevará a la primera versión que guardamos de nuestro archivo, tal y como indica la pantalla con el mensaje: `"HEAD, is now at 342545b primero commit de index."`