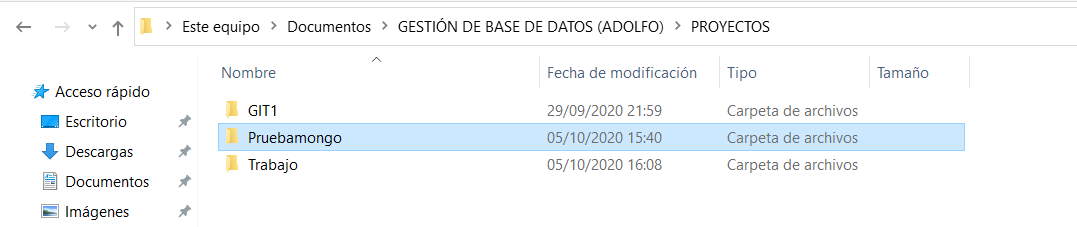
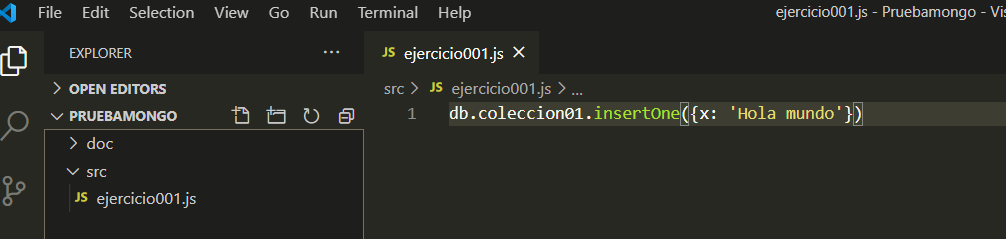
## **PRIMER REPOSITORIO CON MONGODB EN 10 PASOS.**

**PASO 1.**

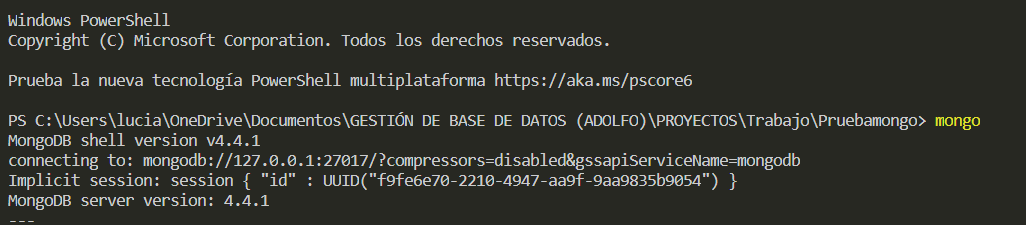
Creamos una nueva carpeta.



A continuación, abrimos Visual Studio Code y cargamos dicha carpeta. Lo siguiente que haremos será crear dos subcarpetas. A la primera le pondremos el nombre ‘’doc’’ y a la otra ‘’src’’. Dentro de la carpeta src crearemos un nuevo archivo con el nombre de nuestro ejercicio terminado en “.js”. En el panel principal escribiremos en la Línea 1 db.coleccion01.insertOne({x: ‘Hola mundo’}) y para salvarlo hacemos CONTROL + S.

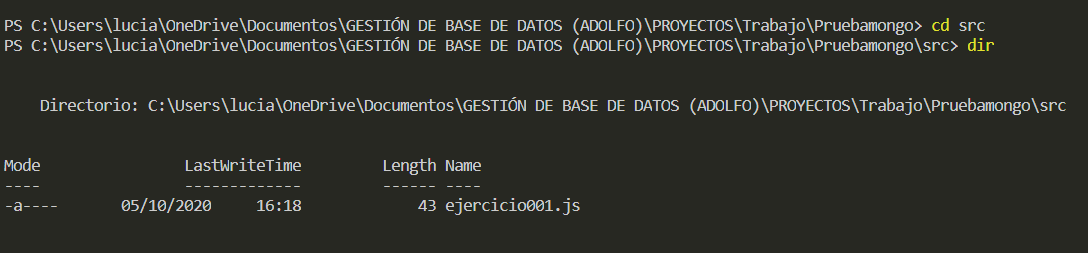


**PASO 2:** abrimos un nuevo terminal y escribimos mongo para saber si tenemos bien instalado el programa.

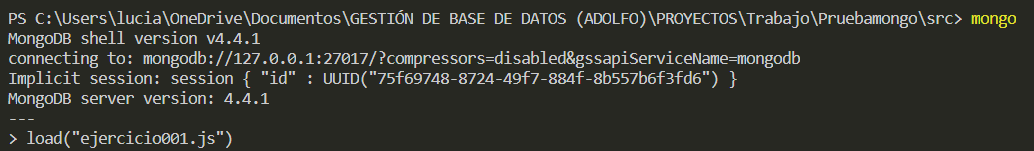


Escribimos cls para dejar la página vacía y empezar a trabajar por lo que tendremos que poner exit para cerrar todas las actividades anteriores.

**PASO 3:** para lanzar este fichero a la carpeta src, escribiremos dc src + dir.

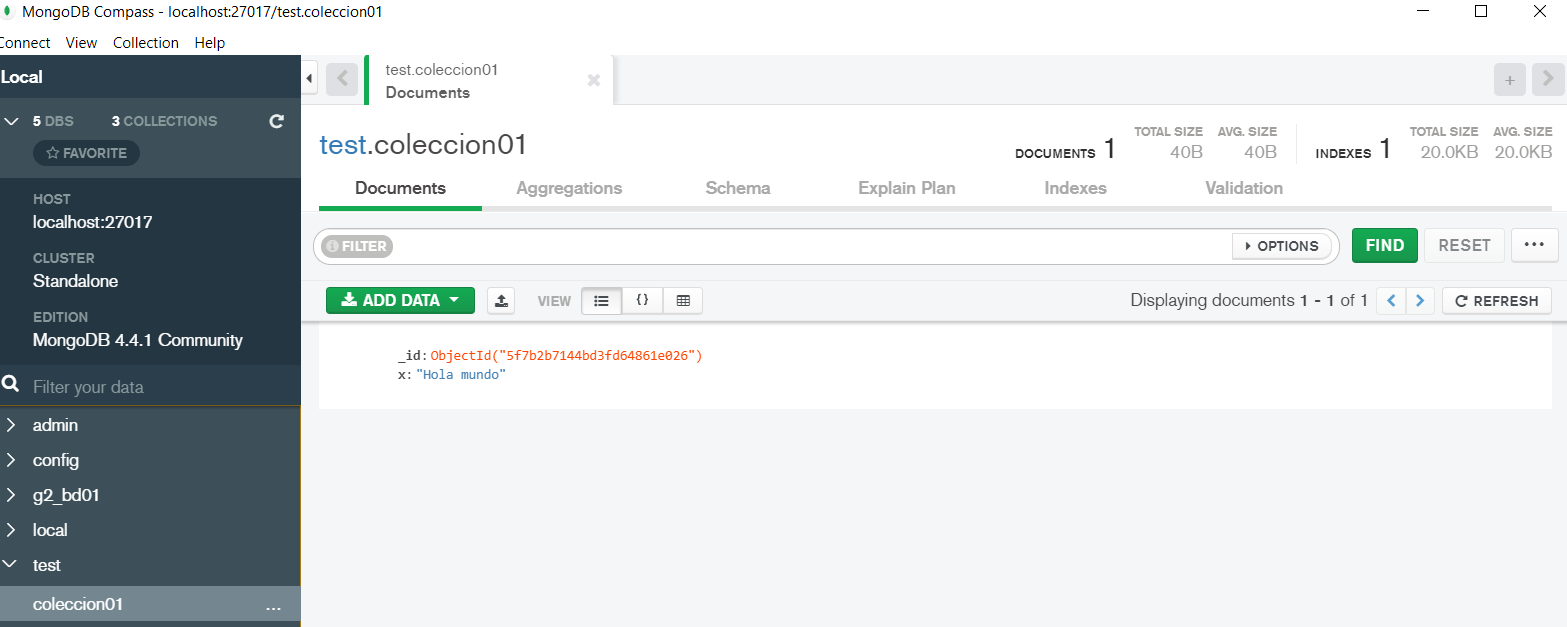


**PASO 4:** vamos a cargar el fichero en Mongo asique para eso primero debemos abrir el comando mongo de nuevo, a continuación, load(“ejercicio001.js”).

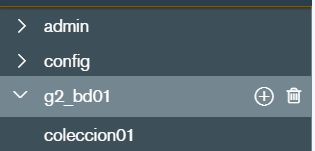


**PASO 5:** ya tenemos el archivo cargado en nuestra base de datos en Mongo, lo comprobaremos entrando en Mongo Compass, abrimos la aplicación y le damos a ‘CONNECT’.

Una vez dentro buscamos la base de datos que ha sido creada de forma predeterminada por Mongo llamada ‘’test’’. Esto ocurre porque no le pusimos a que base de datos queremos mandar el archivo.



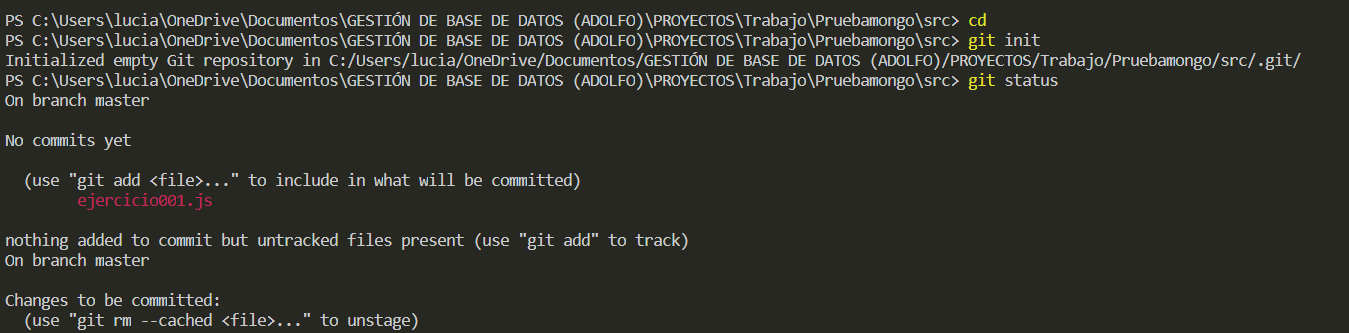
**PASO 6:** para mandar nuestro archivo a la base de datos que nosotros queremos escribiremos en el terminal: use g2\_bd01. Si lanzamos de nuevo este fichero ahora db es la base de datos a la que mandaremos nuestro archivo para ello, debemos cargarlo de nuevo: load(“ejercicio001.js”)

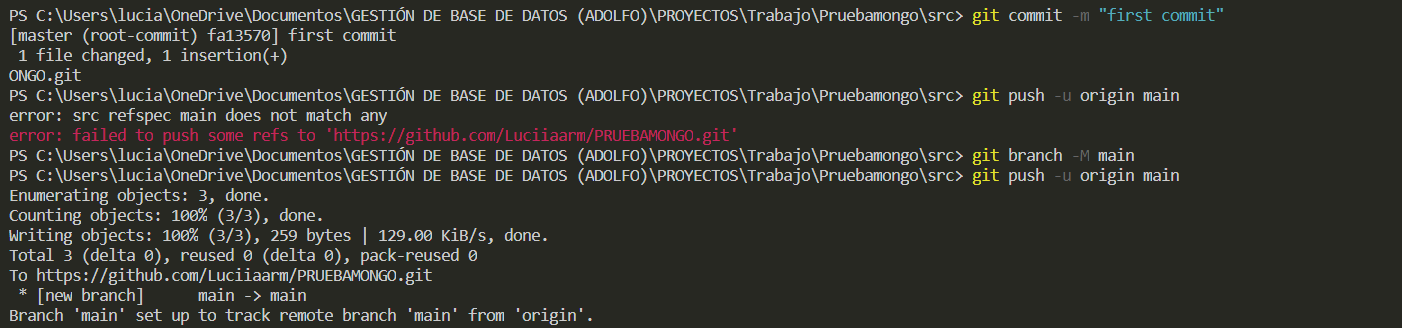
**PASO 7:** una vez que tengamos ya nuestra bd\_01, escribimos en nuestro terminal de Visual Studio Code, exit y damos por finalizada la carga del archivo a la base de datos. Ahora lo que debemos hacer es mandar el archivo ejercicio001.js a la nube de GitHub.

Accedemos a nuestra carpeta src con el comando cd. Luego iremos a la página de GitHub y crearemos un nuevo repositorio con el nombre de nuestra carpeta principal, en mi caso, PRUEBAMONGO.

Creamos una carpeta local en el repositorio con el comando git init, aunque podemos observar si ponemos git status que la carpeta está por crear y preparada para añadir la información.



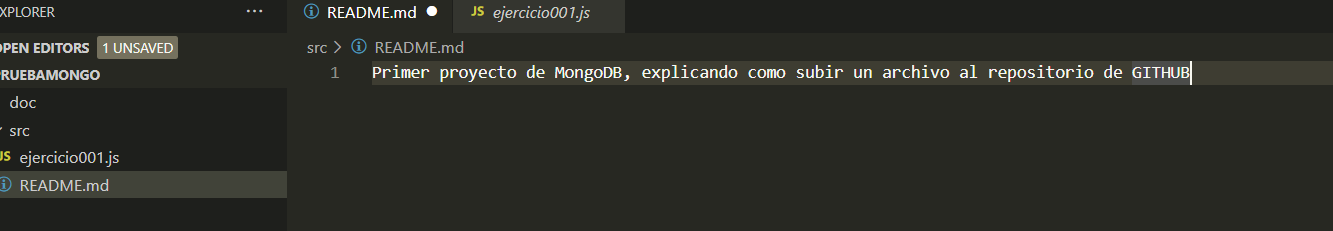
**PASO 8**: añadimos el contenido de la carpeta src con el comando git add . una vez echo los ficheros ya están preparados para hacer commit, git commit -m "first commit". Una vez cargados ya hacemos el push y los mandamos al repositorio finalmente, git push -u origin main, si nos da algún tipo de error con el origin main, debemos poner primero el comando git branch -M main y luego repetir git push -u origin main

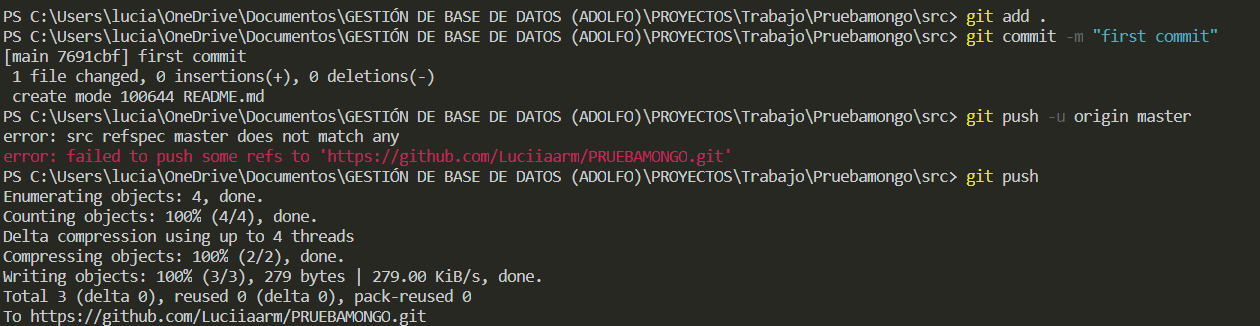


**PASO 9: CREACIÓN DE UN README.**

Creamos un nuevo archivo dentro de la carpeta SRC, con el nombre ‘’README.md’’. A continuación, a la derecha escribiremos lo que queremos que ponga en nuestro readme, preferentemente debe aparecer de que se trata el repositorio, que archivos puedes encontrar y cuales son sus utilidades. Para ello después de haber escrito lo que queremos explicar y recalcar del trabajo, solo tenemos que hacer los siguientes comandos en orden:

1. Git add.
2. Git commit -m ‘’first commit’’’
3. Git push -u origin master (en mi caso este comando me dio un fallo como se puede ver abajo, lo único que hay que hacer para solucionarlo es dejar solo el comando git push)



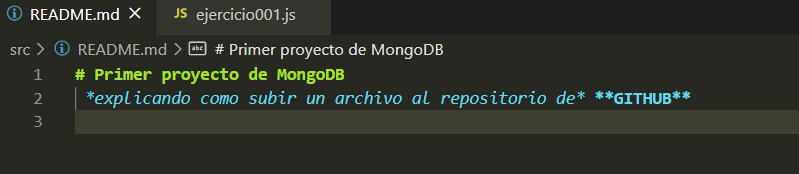


**PASO 10**: personalización de las letras del archivo README.md

PARA PONER UN TÍTULO: escribimos antes del texto #

PARA PONER LA LETRA EN NEGRITA: escribimos antes y después del texto que queremos resaltar \*\*

PARA PONER LA LETRA EN CURSVIA: escribimos antes y después del texto que queremos resaltar \*



Una vez escrito, CONTROL + S y cargarlo de nuevo en GitHub.

1. Git add.
2. Git commit -m ‘’first commit’’’
3. Git push