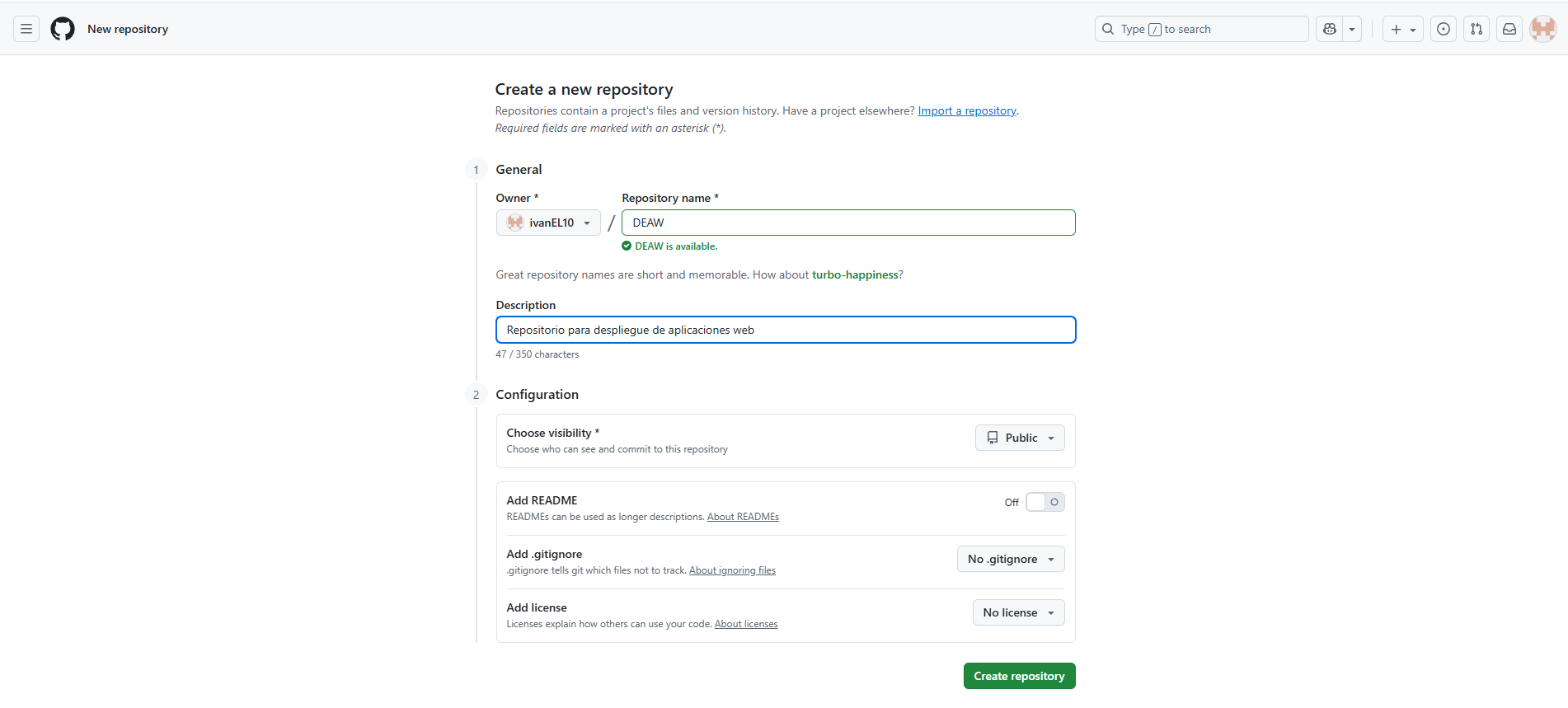
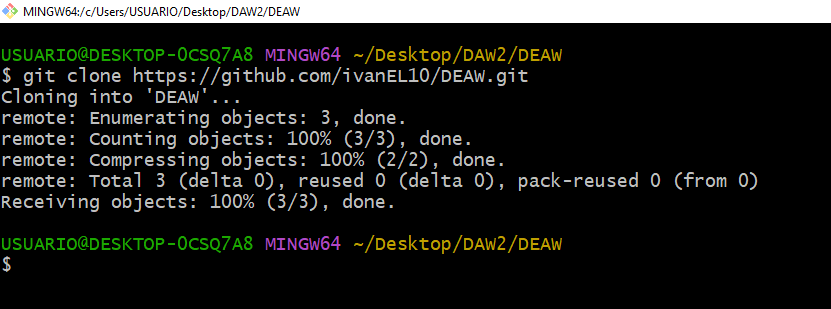
**Enunciado ejercicios Git y GitHub**

Repositorio DEAW

1. Crear un repositorio en vuestro GitHub llamado **DEAW**.



1. Clonar vuestro repositorio en local.



README

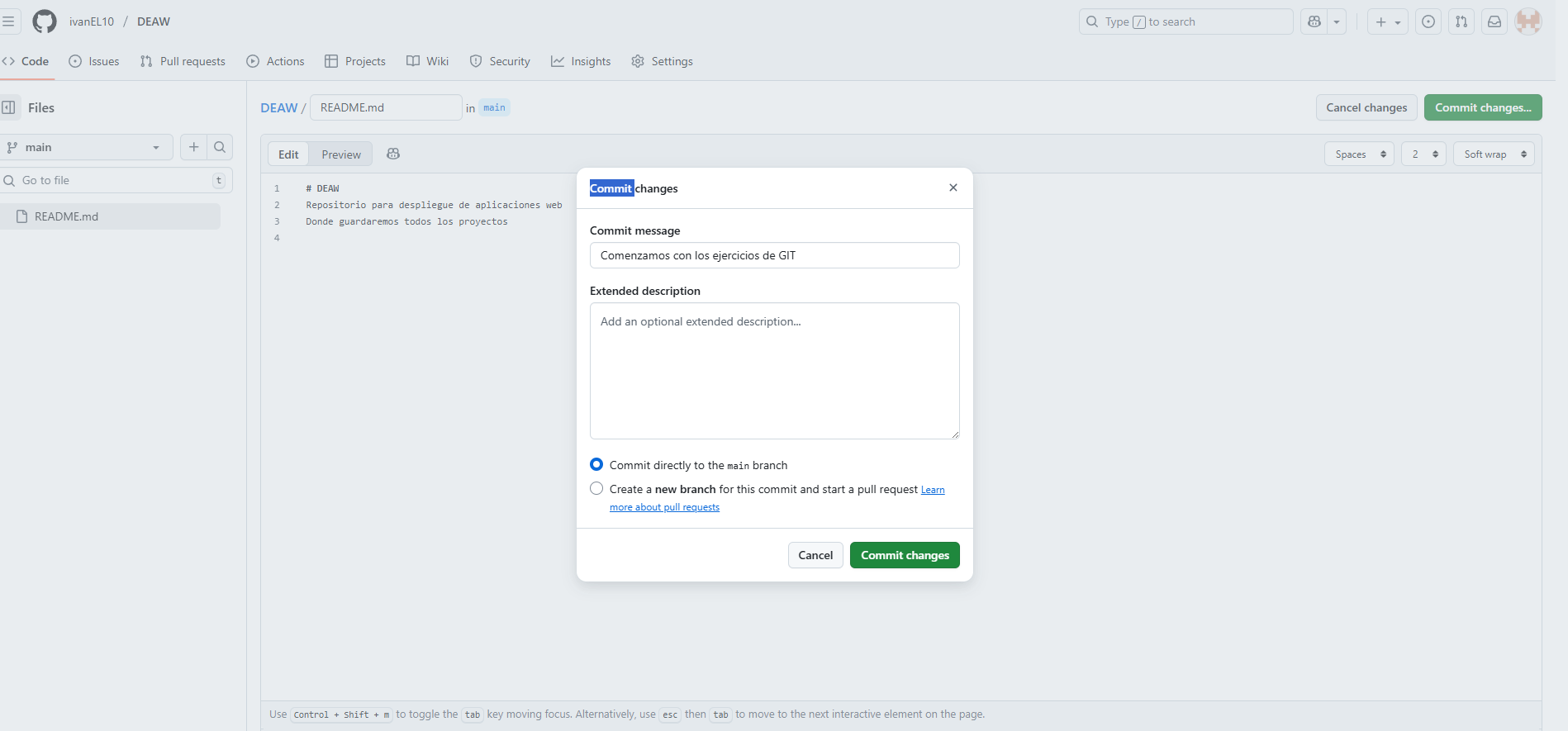
1. Crear (si no lo habéis creado ya) en vuestro repositorio local un documento **README.md**.

**Nota**

Escribir un pequeño texto en este README a propósito del repositorio y el módulo para el que se utilizará·

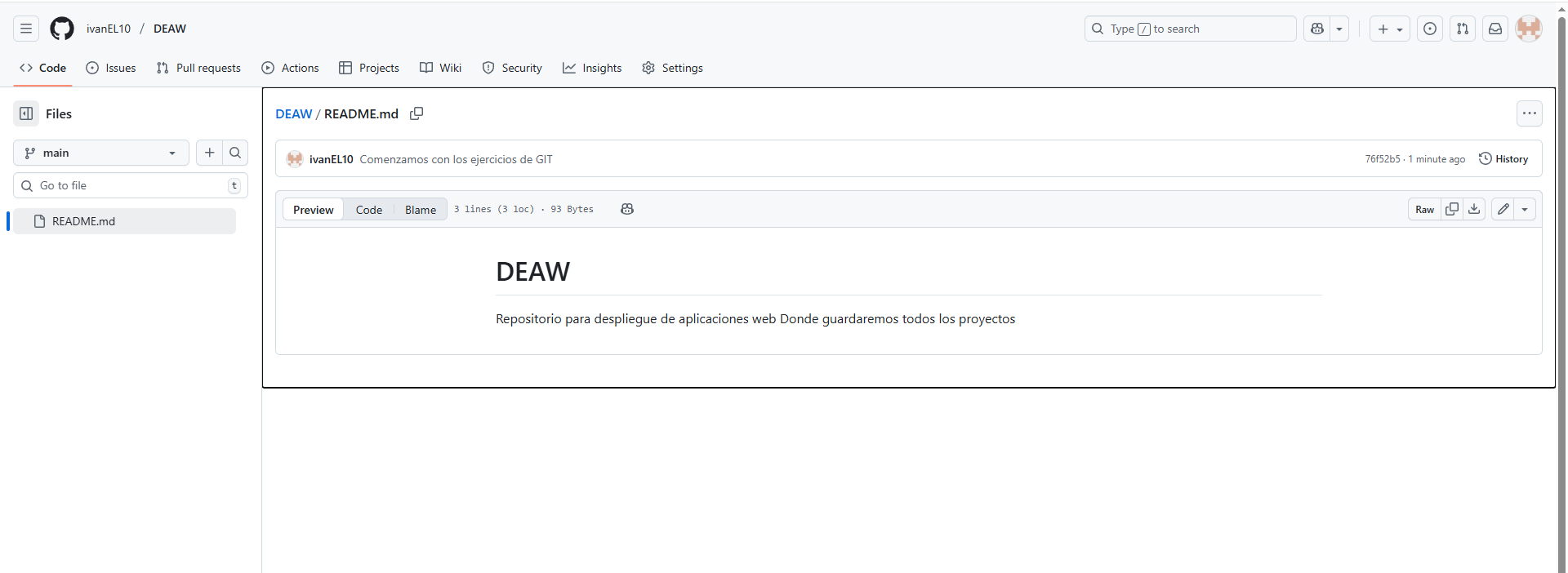
Commit inicial

1. Realizar un commit inicial con el comentario Comenzamos con los ejercicios de Git



Push inicial

1. Subir los cambios al repositorio remoto.



Cambios hechos en el repositorio remoto

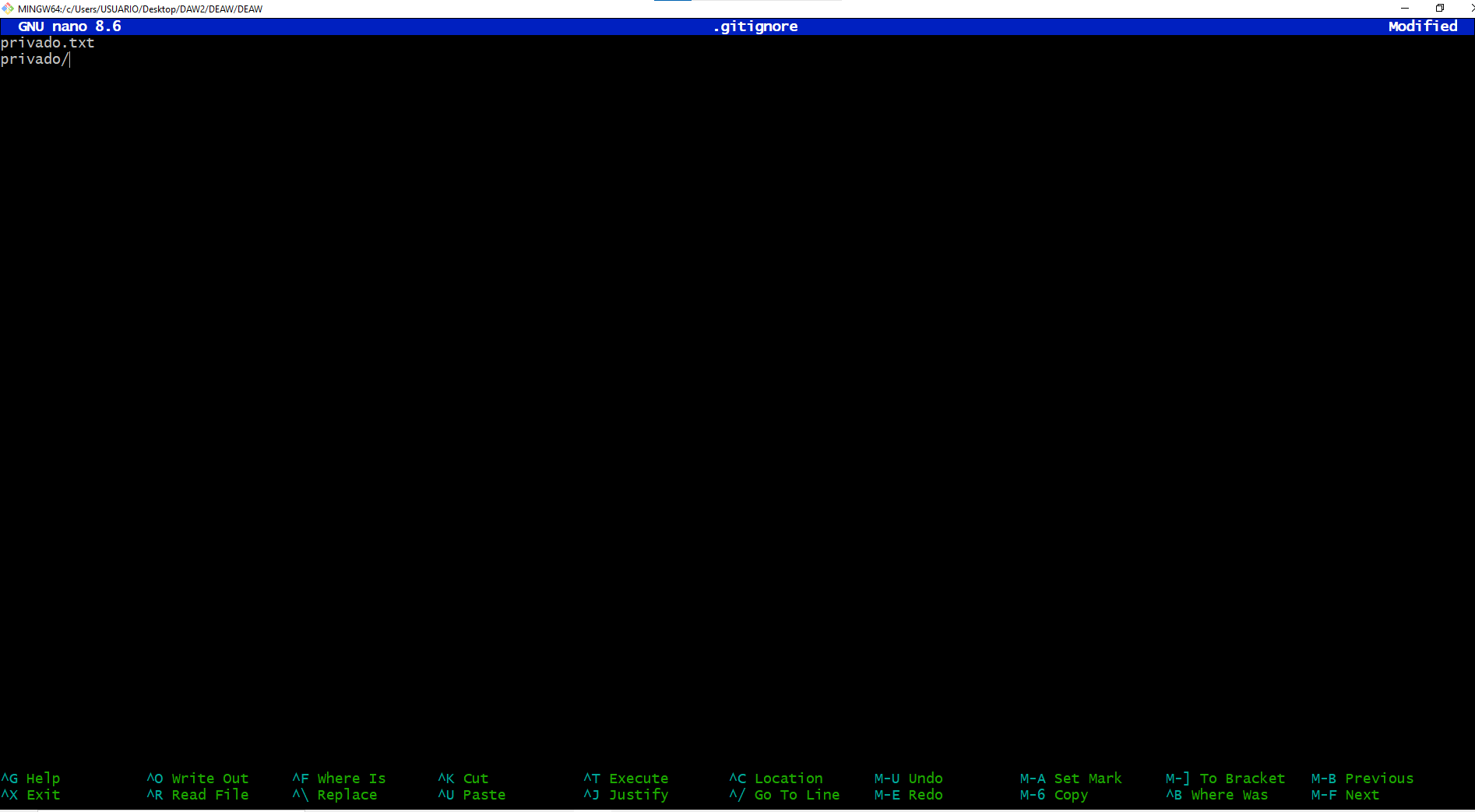
1. Ignorar archivos
2. Crear en el repositorio local un fichero llamado **privado.txt**.
3. Crear en el repositorio local una carpeta llamada **privada**.



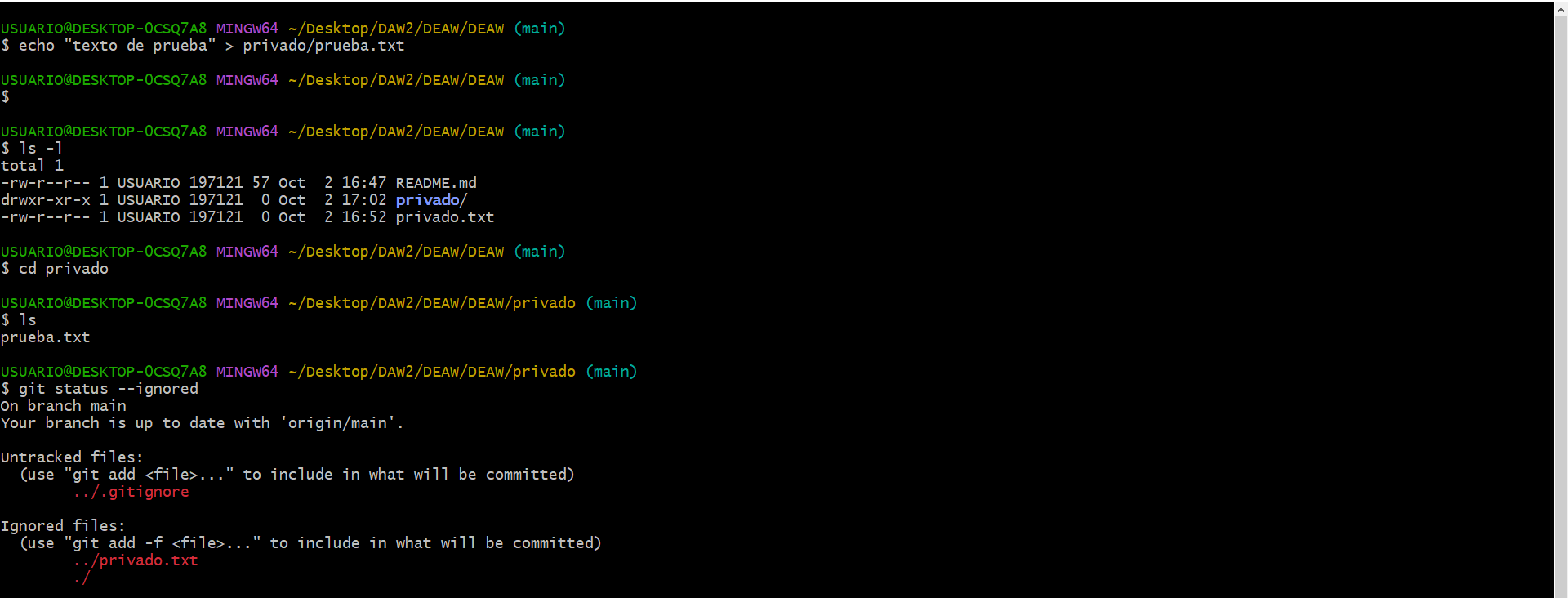
1. Realizar los cambios oportunos para que tanto el archivo como la carpeta sean ignorados por git.



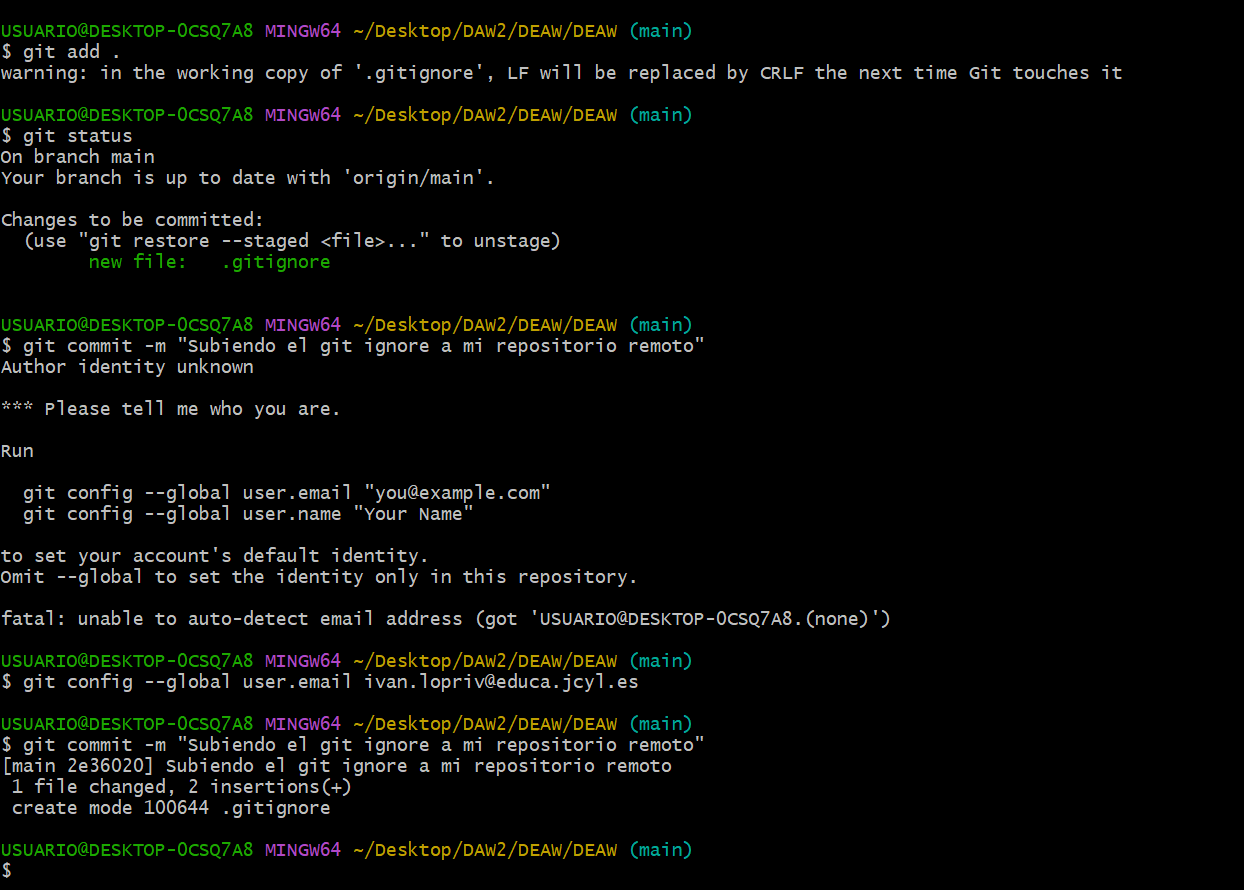
Creo la carpeta .gitignore para ignorar los siguientes archivos y lo abro con el editor de texto nano



Añado estos 2 archivos a esta “lista” para que pueda ignorarlo



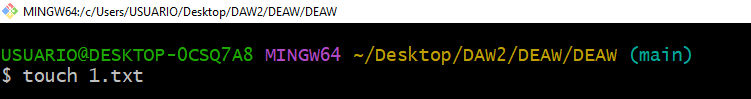
He creado un archivo de prueba dentro de la carpeta para comprobar que esta ignorada, al ejecutar el comando git status –ignored aparecen todos los archivos ignorados, en este caso aparece privado.txt y ./ , este último significa que estoy dentro de la propia carpeta privado por lo que si vuelvo a la original aparece correctamente privado/



Hago un commit subiendo todos los cambios que he realizado haciendo este ejercicio. Tuve que introducir el correo de mi cuenta de github para que me dejara realizar el commit ya que acabo de instalar gitbash en el ordenador de mi casa

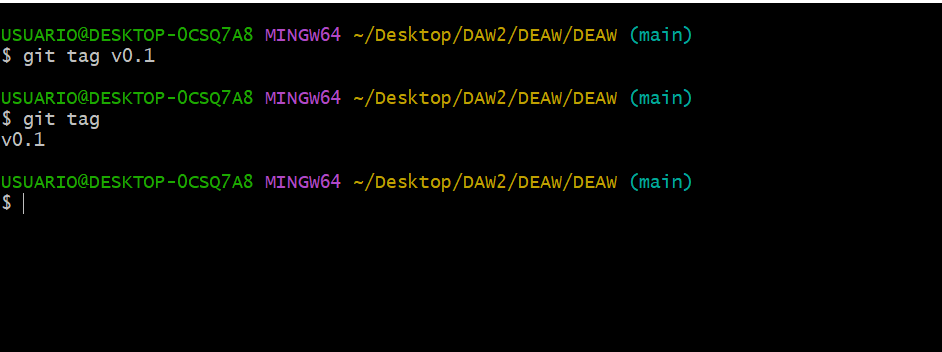
Añadir fichero 1.txt

1. Añadir fichero **1.txt** al repositorio local.



Crear el tag v0.1

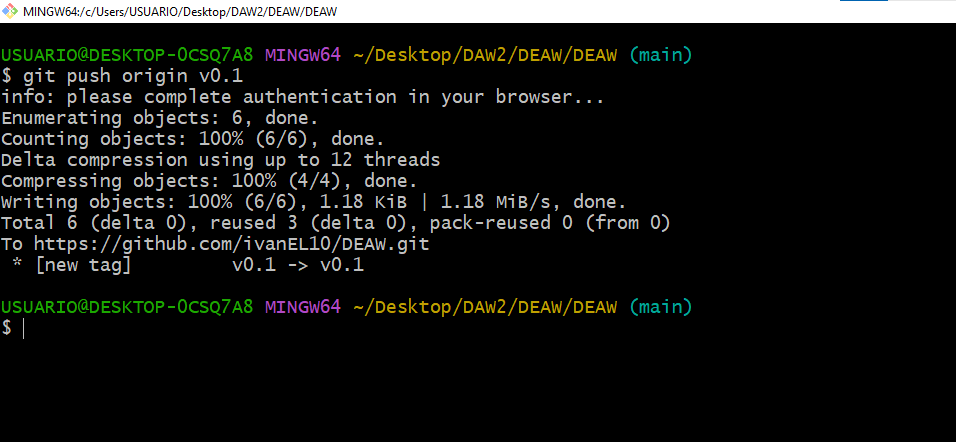
1. Crear un tag **v0.1**.

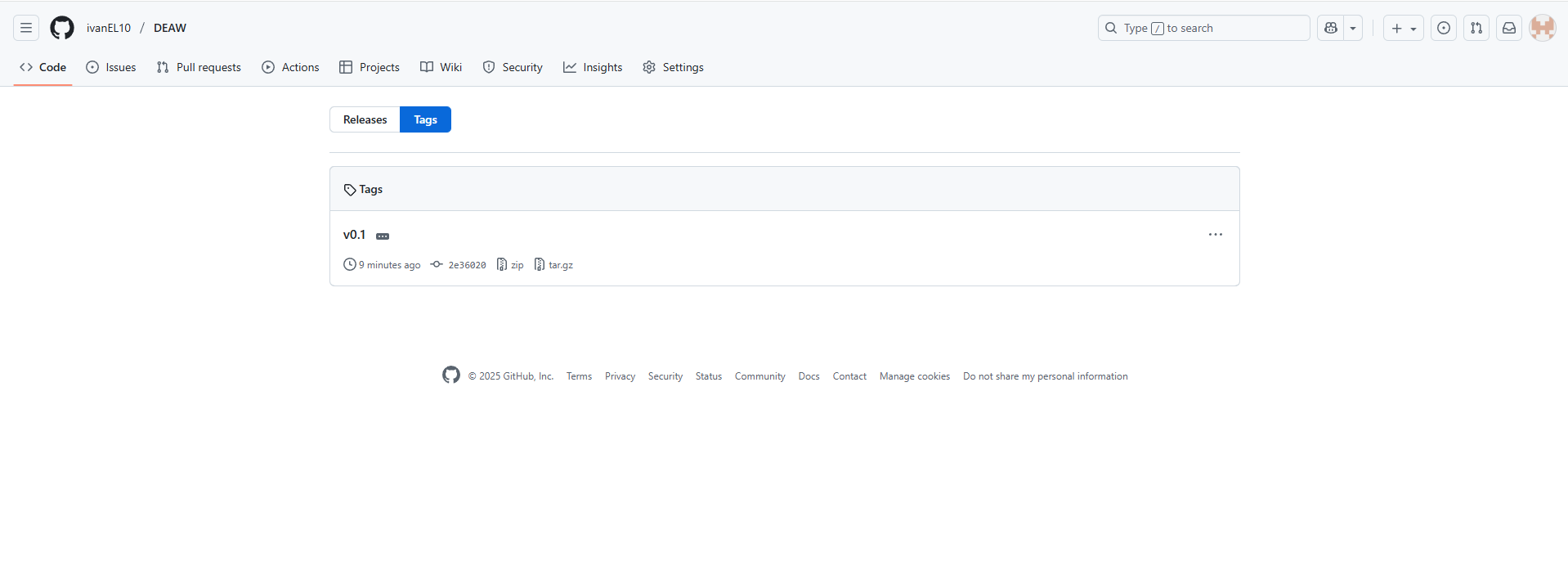


Creo el tag y verifico que existe

Subir el tag v0.1

1. Subir los cambios al repositorio remoto.

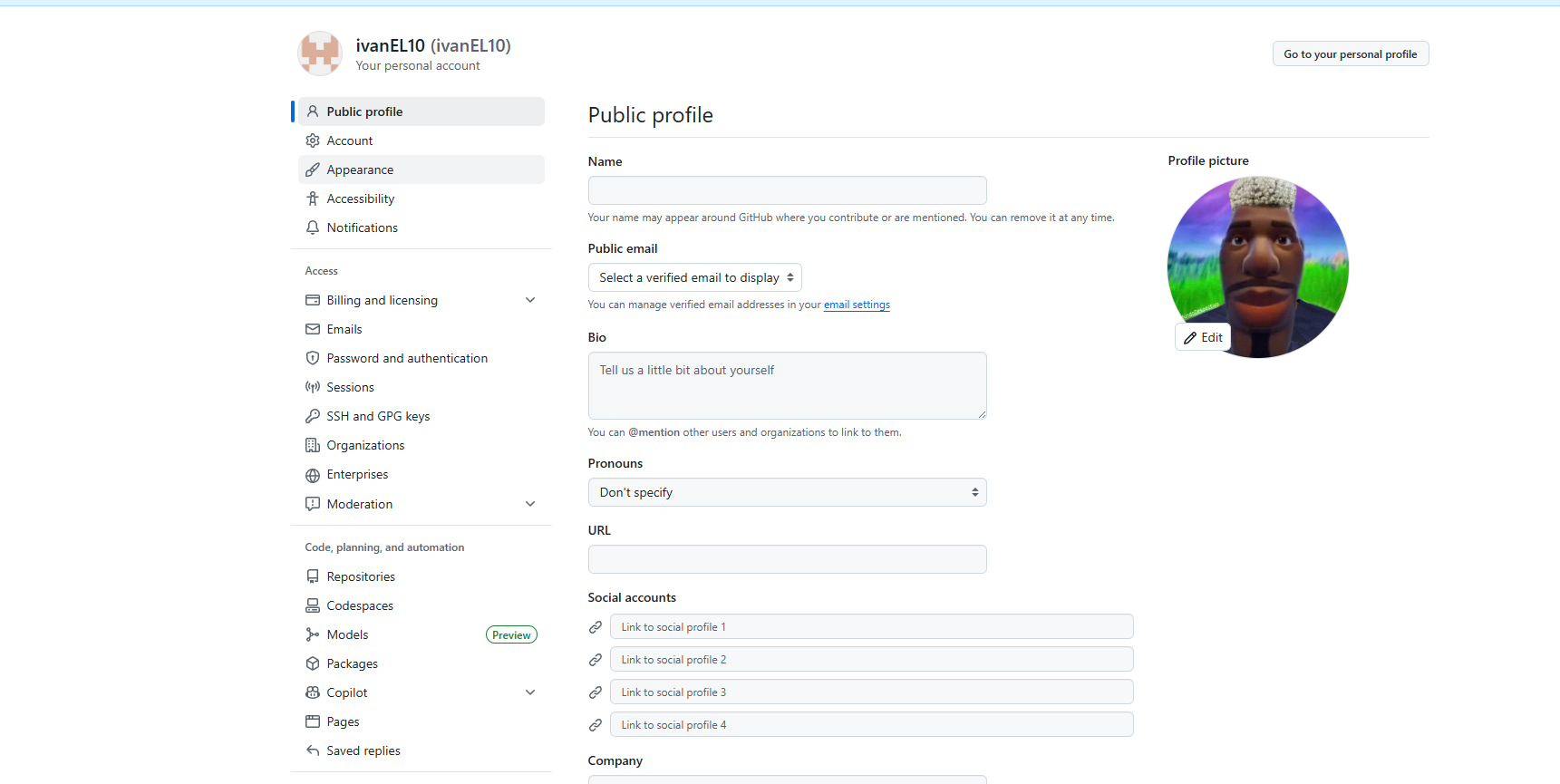




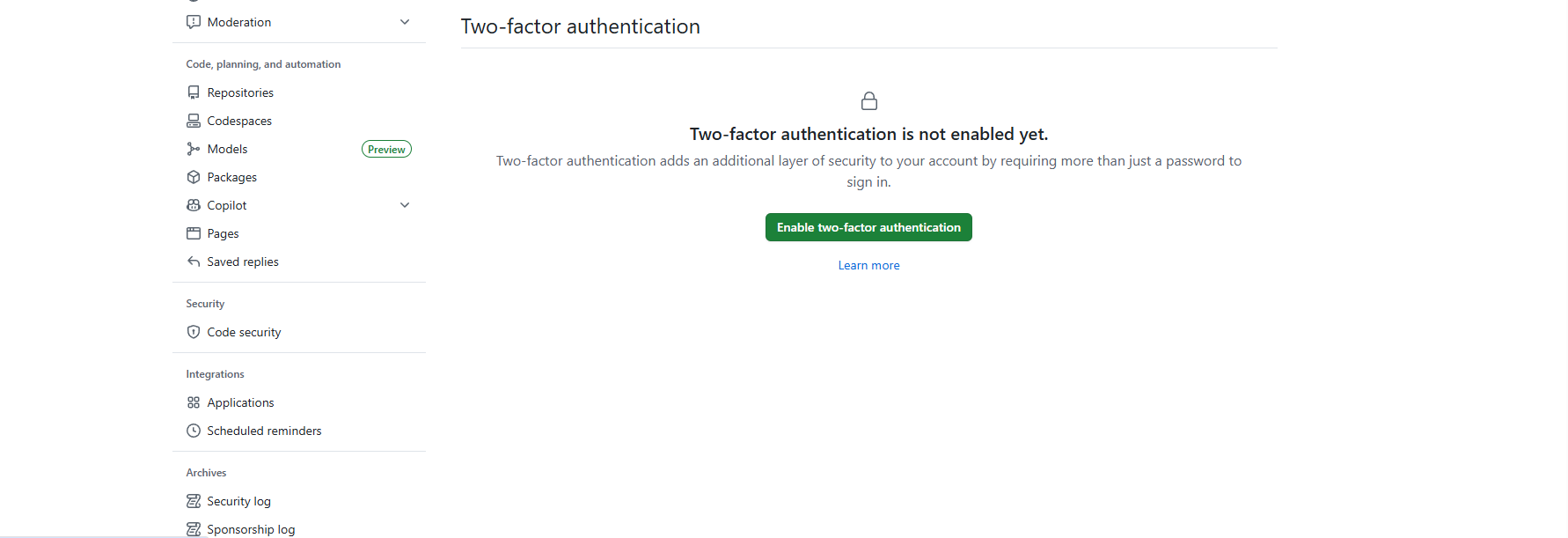
Comprobación del tag de que se ha subido correctamente

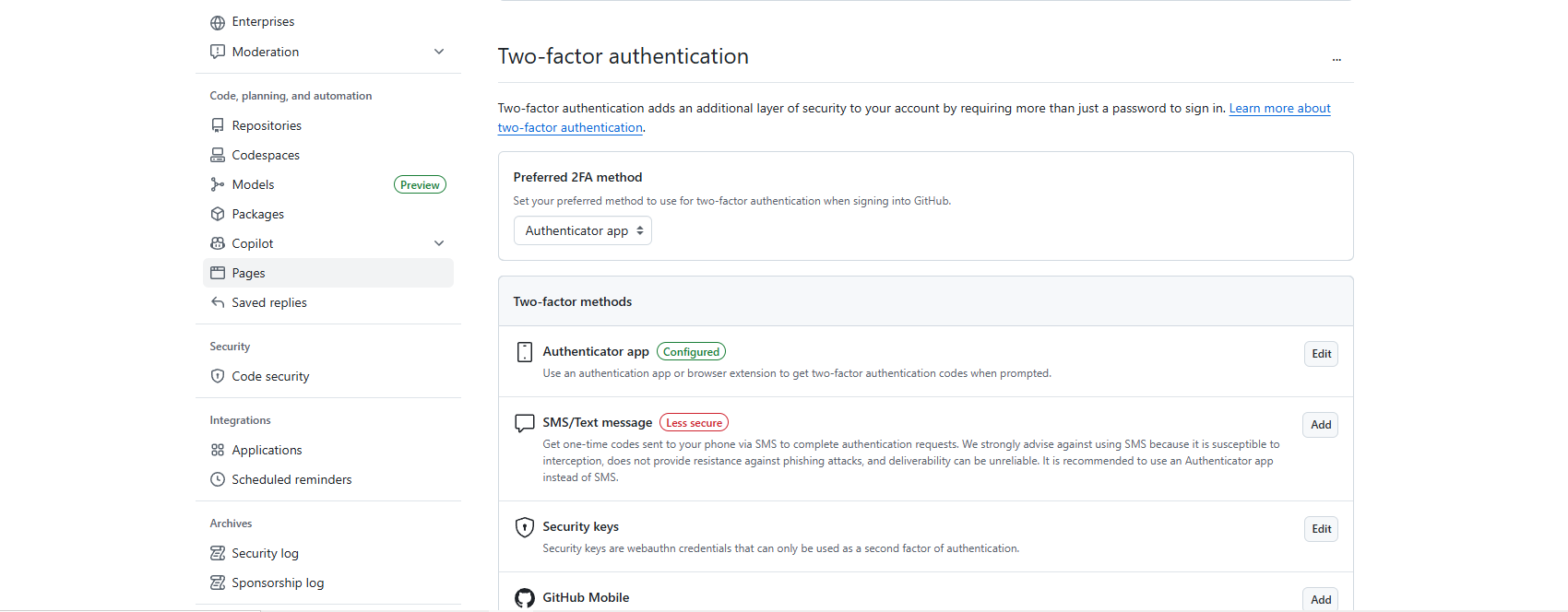
Cuenta de GitHub

1. Poner una foto en vuestro perfil de GitHub.



1. Poner el doble factor de autentificación en vuestra cuenta de GitHub.

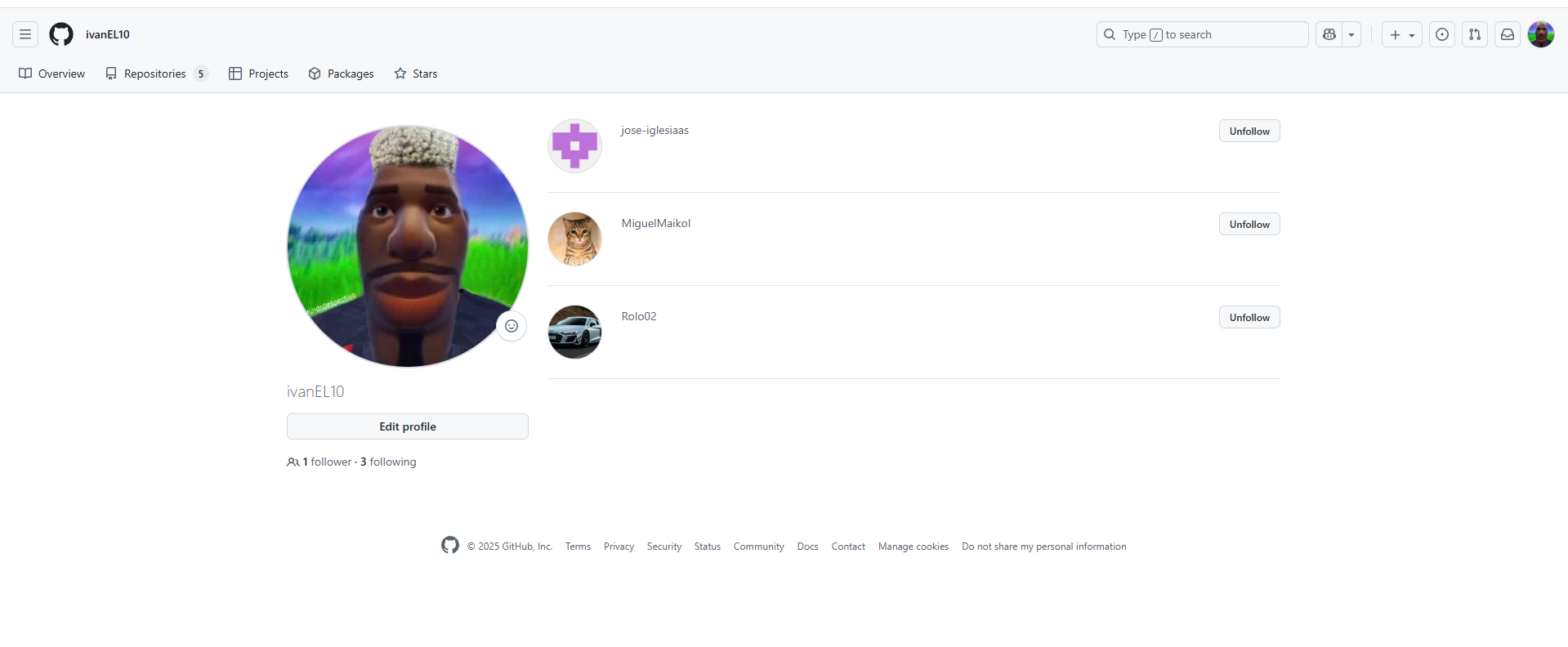




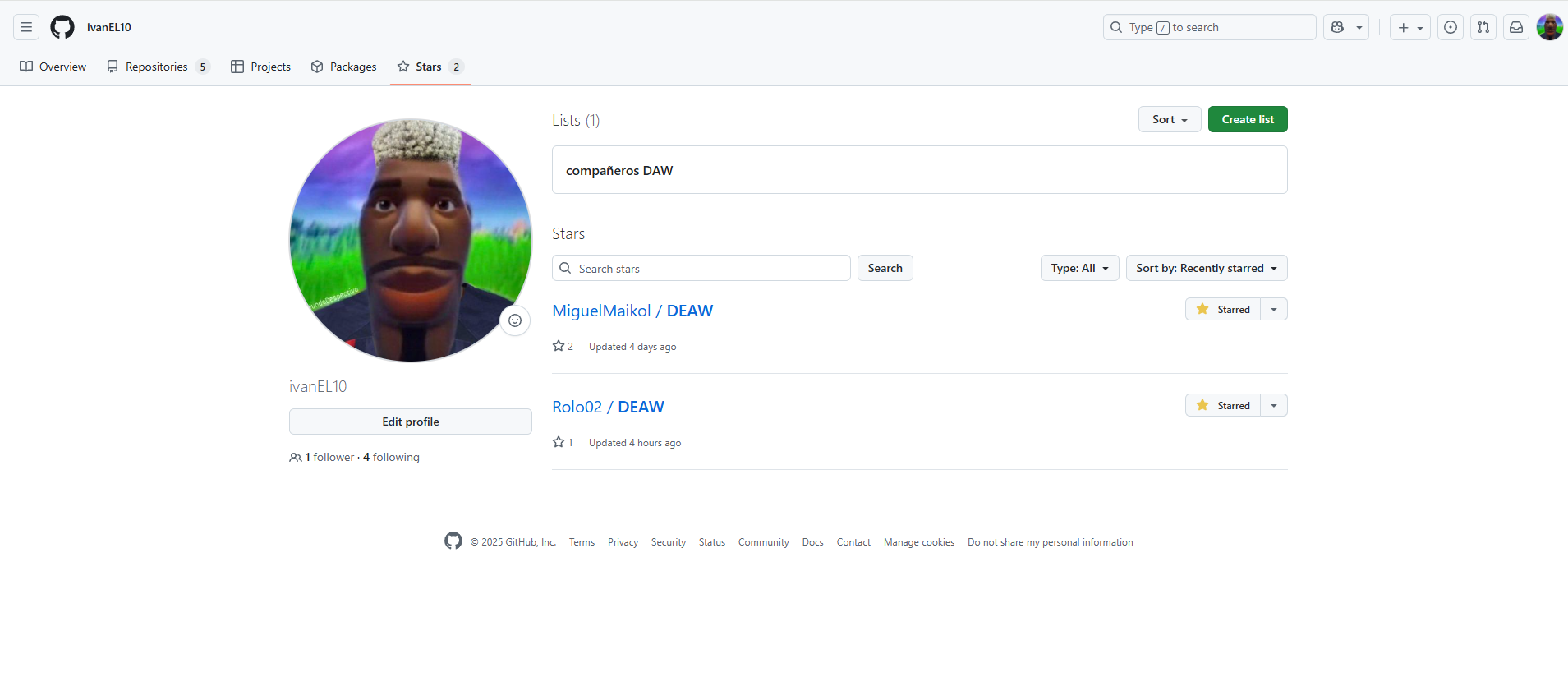
Autentificación ya realizada

Uso social de GitHub

1. Preguntar los nombres de usuario de GitHub de 2 de tus compañeros de clase, búscalos, y sígueles.



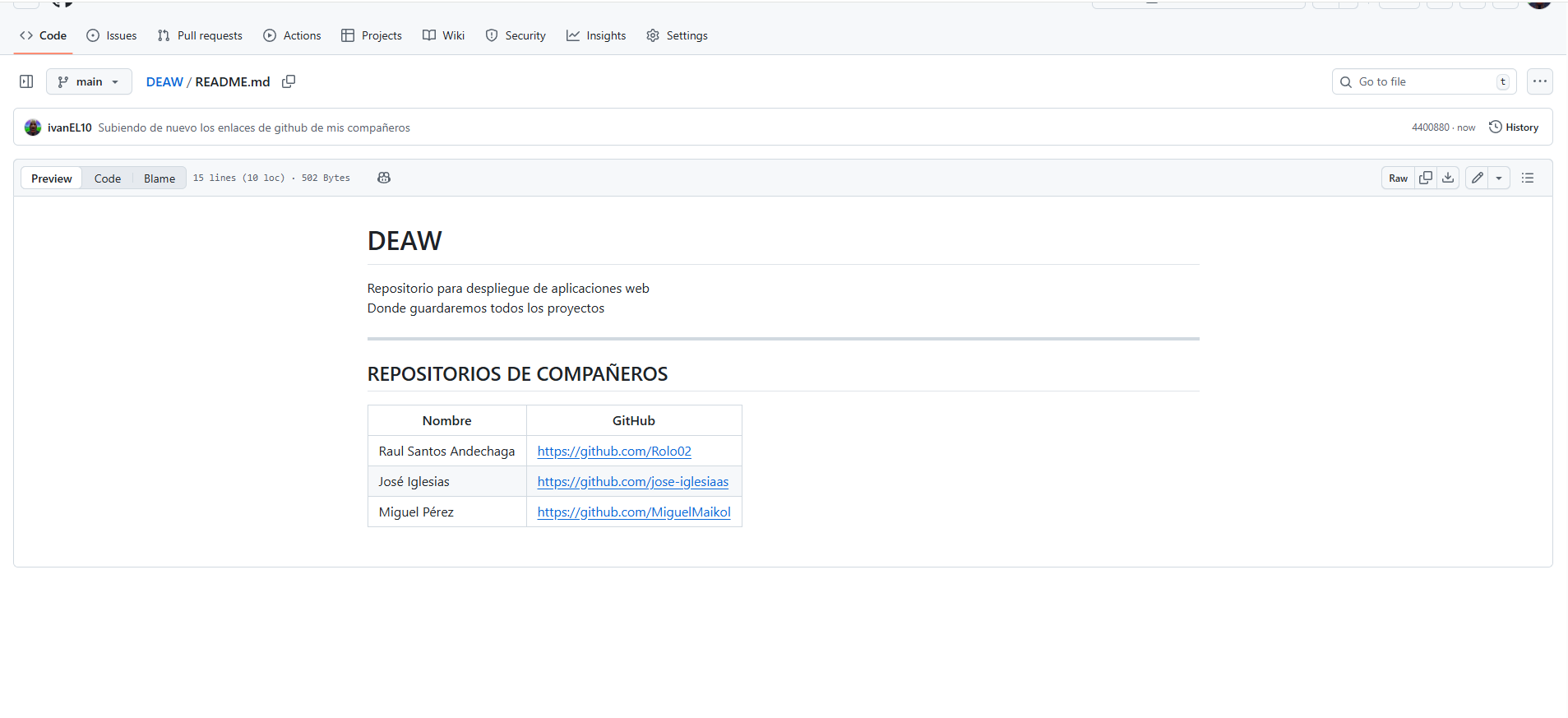
1. Seguir los repositorios **DEAW** del resto de tus compañeros.
2. Añadir una estrella a los repositorios **DEAW** del resto de tus compañeros.



Crear una tabla

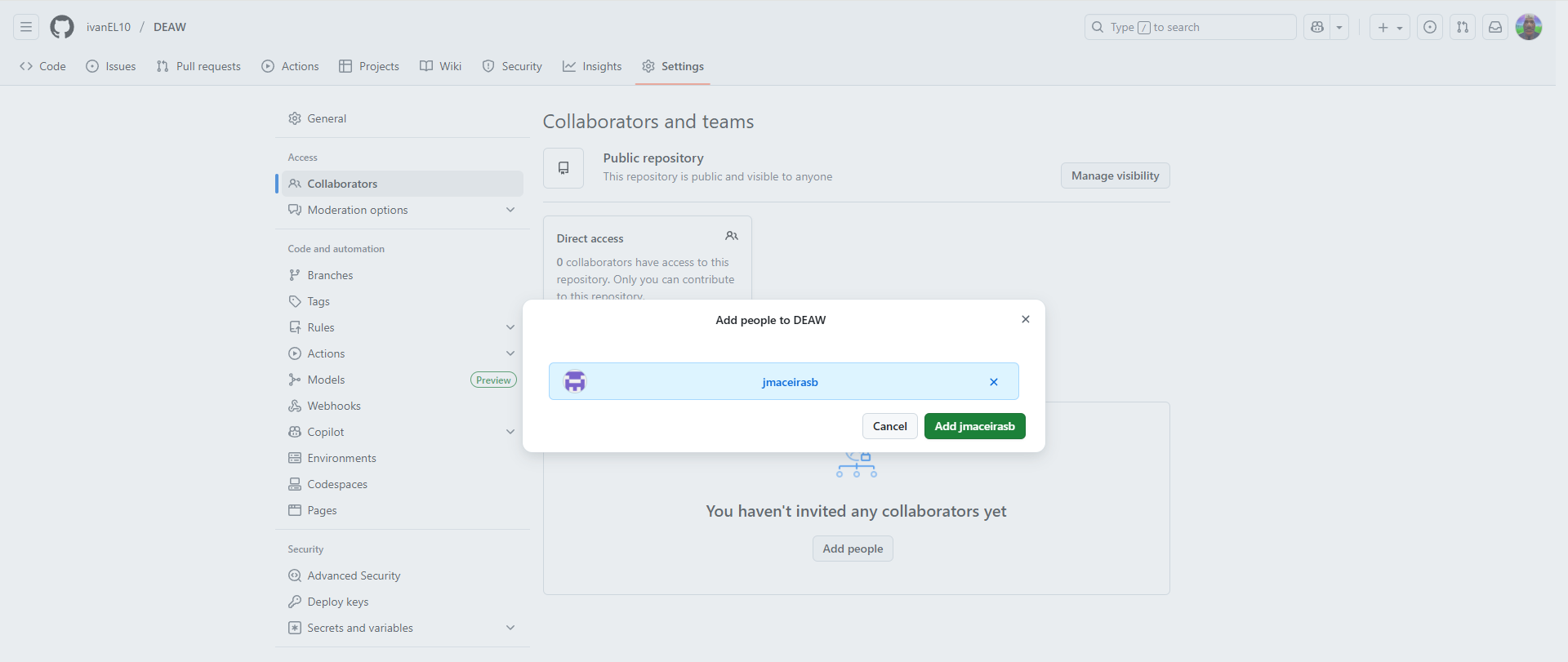
1. Crear una tabla de este estilo en el fichero **README.md** con la información de varios de tus compañeros de clase:

| **NOMBRE** | **GITHUB** |
| --- | --- |
| Nombre del compañero 1 | [enlace a github 1](http://github.com/asanzdiego) |
| Nombre del compañero 2 | [enlace a github 1](http://github.com/asanzdiego) |
| Nombre del compañero 3 | [enlace a github 3](http://github.com/asanzdiego) |



Colaboradores

1. Poner a [github.com/jmaceirasb](https://github.com/jmaceirasb)como colaborador del repositorio DEAW

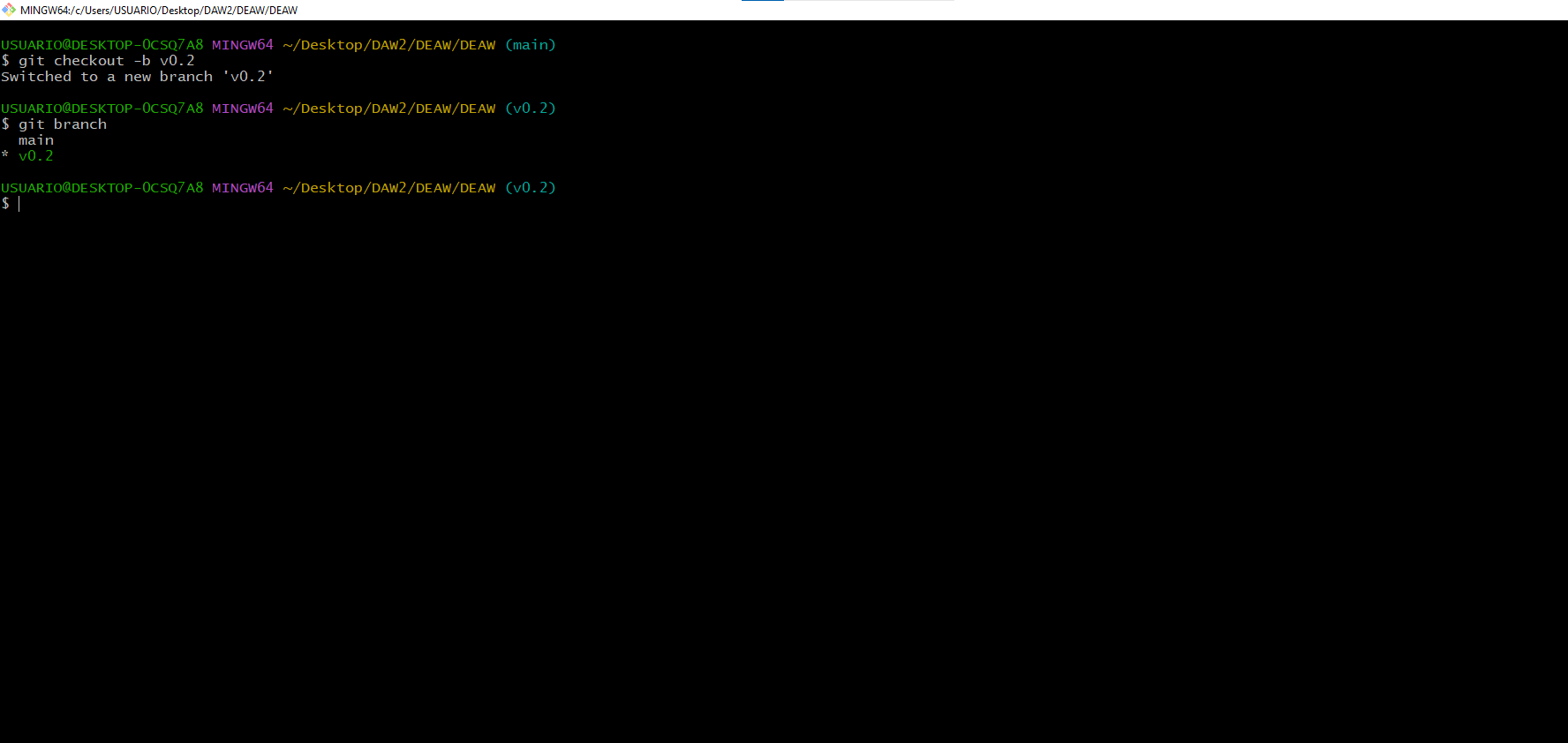


Notas

* Este ejercicio es continuación del anterior por lo que tendréis que seguir trabajando en el repositorio **DEAW**.
* También tendréis que ir poniendo los **comandos** que habéis tenido que utilizar durante todos los ejercicios y las **explicaciones y capturas de pantalla** que consideréis **necesarias** en el informe.

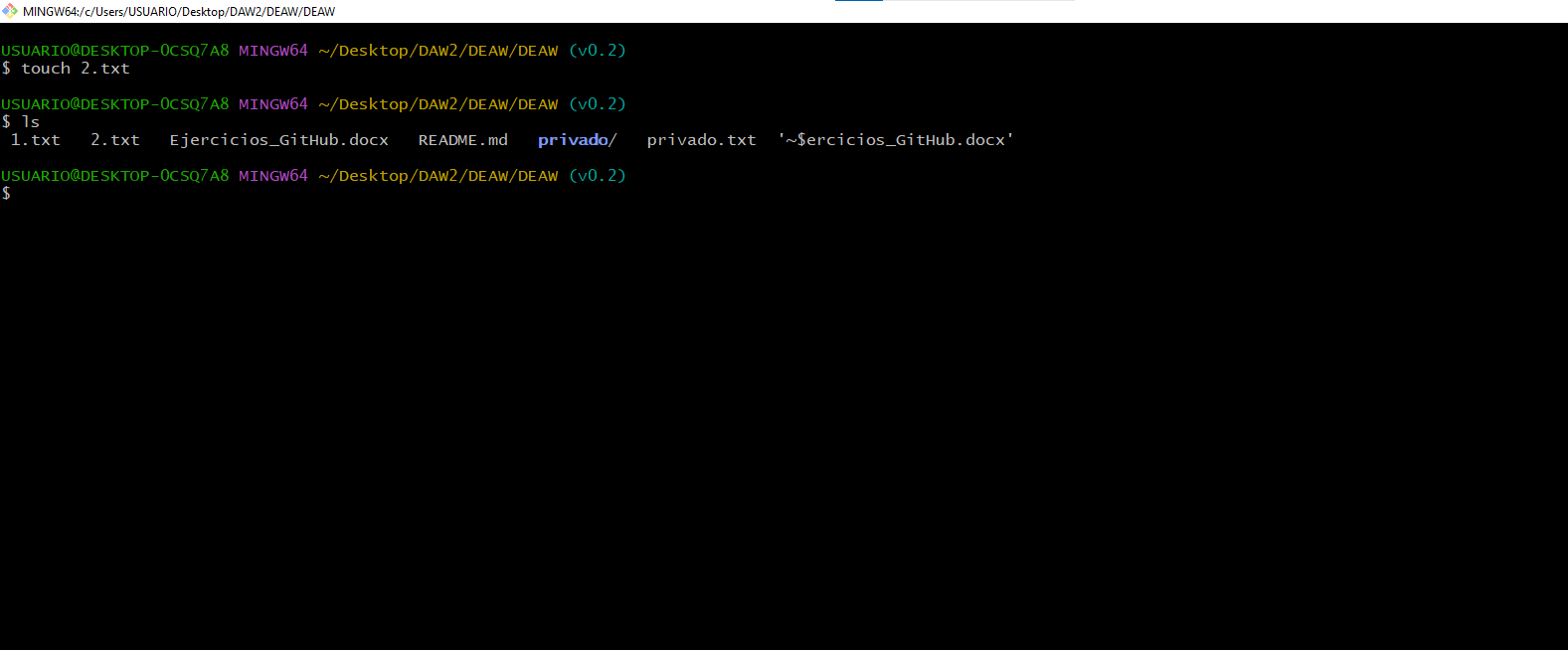
Crear una rama v0.2

1. Crear una rama **v0.2**.
2. Posiciona tu carpeta de trabajo en esta rama.



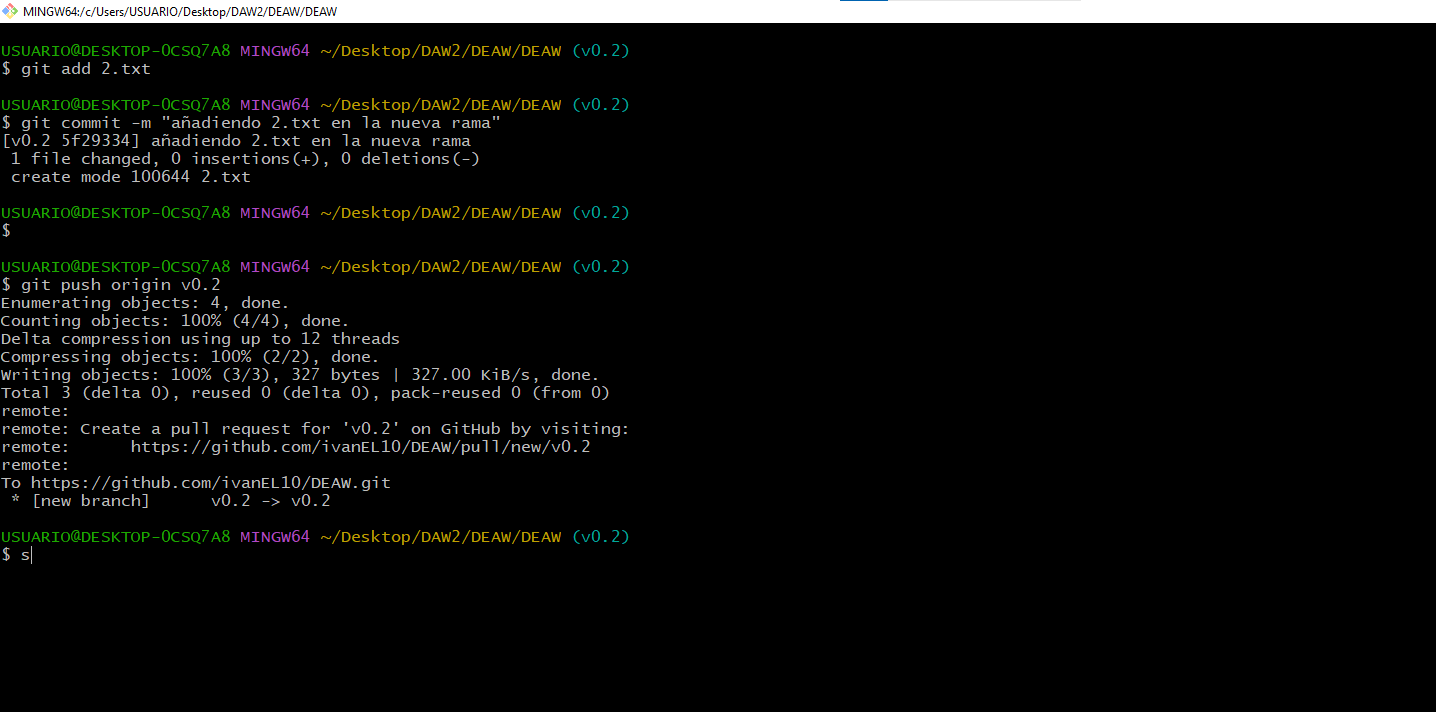
Añadir fichero 2.txt

1. Añadir un fichero **2.txt** en la rama **v0.2**.



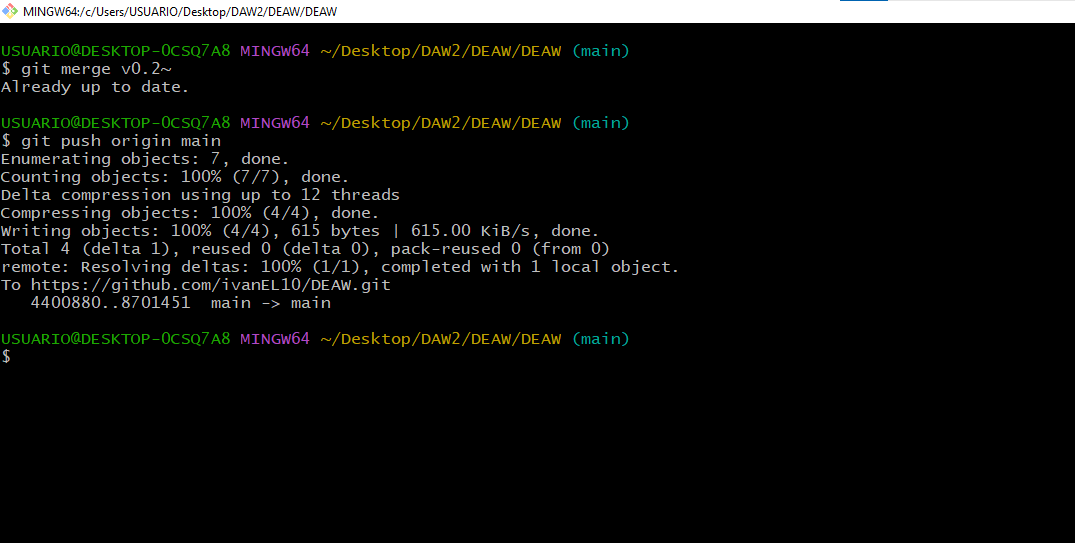
Crear rama remota v0.2

1. Subir los cambios al repositorio remoto.



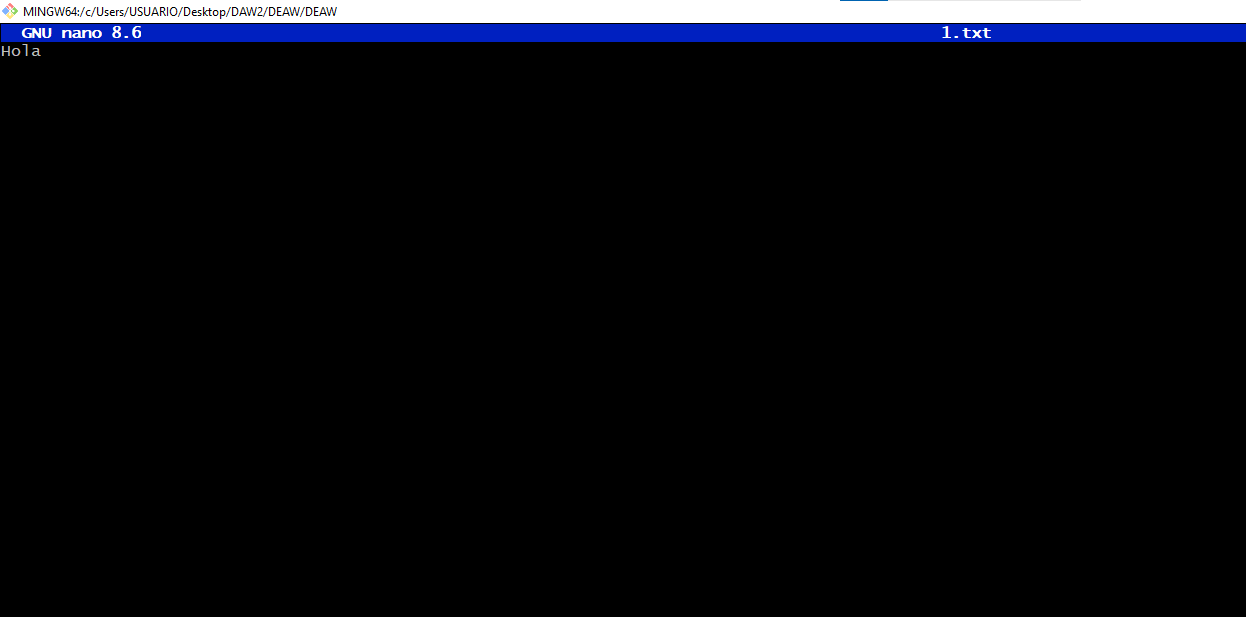
Merge directo

1. Posicionarse en la rama **master**.
2. Hacer un merge de la rama **v0.2** en la rama **master**.



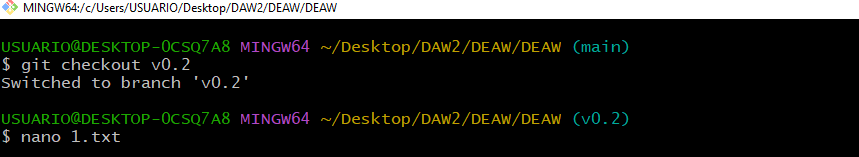
Merge con conflicto

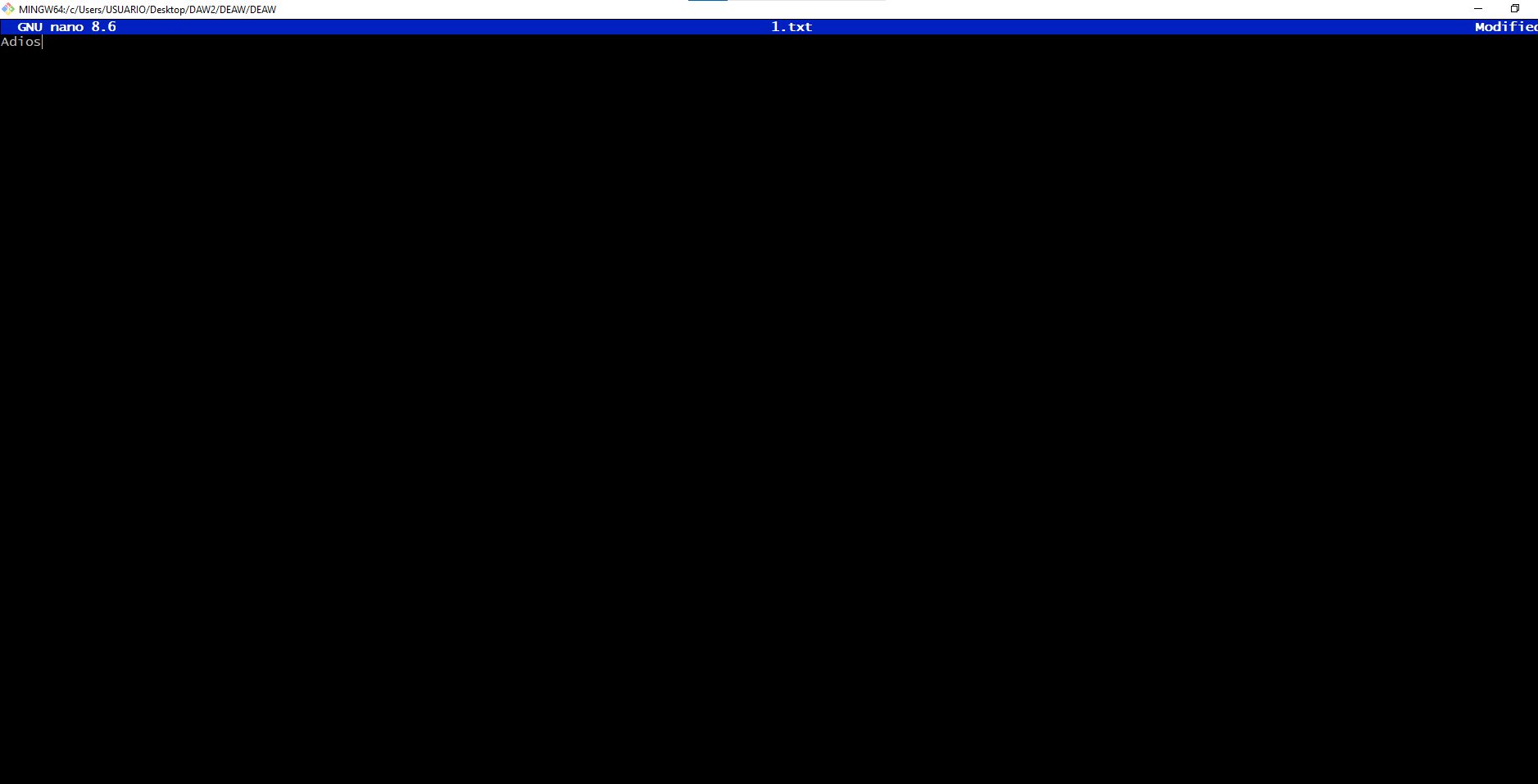
1. En la rama **master** poner **Hola** en el fichero **1.txt** y hacer commit.

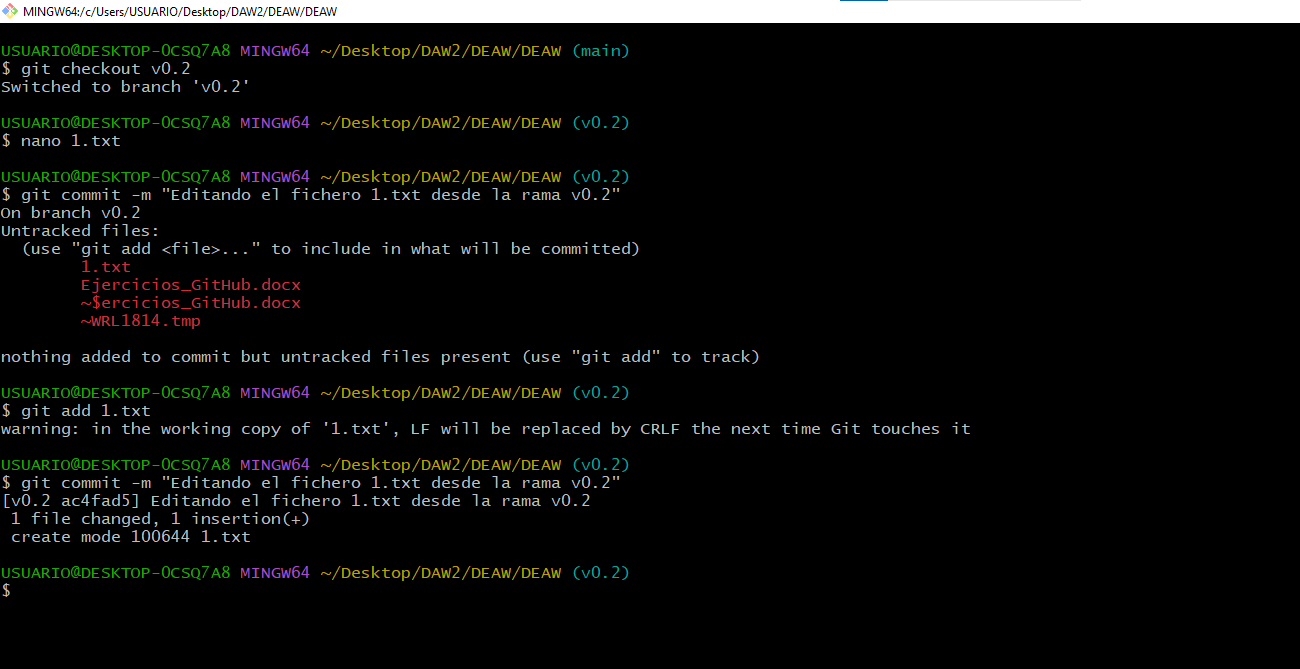




1. Posicionarse en la rama **v0.2** y poner **Adios** en el fichero "1.txt" y hacer commit.

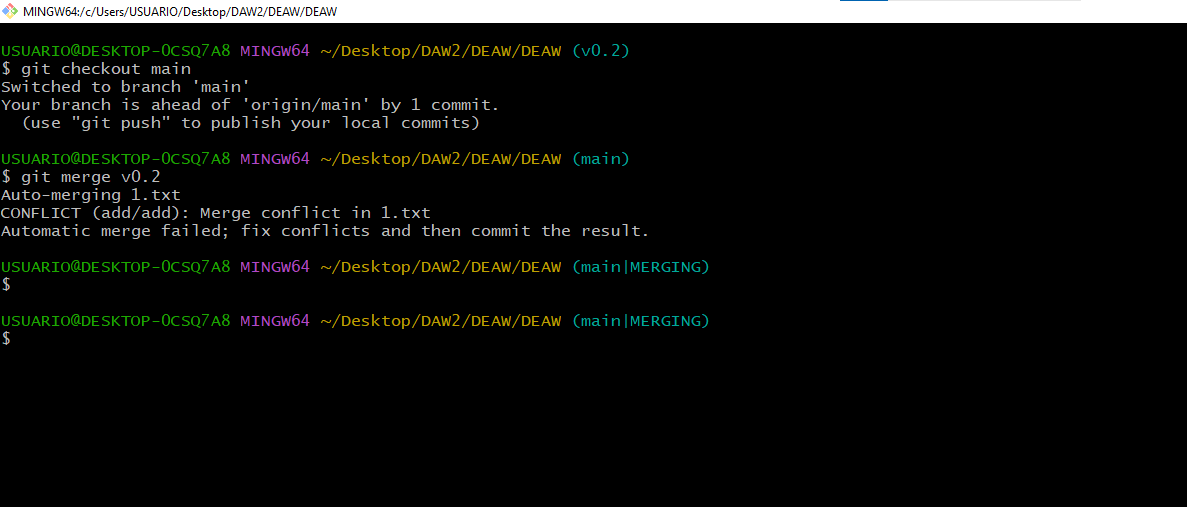


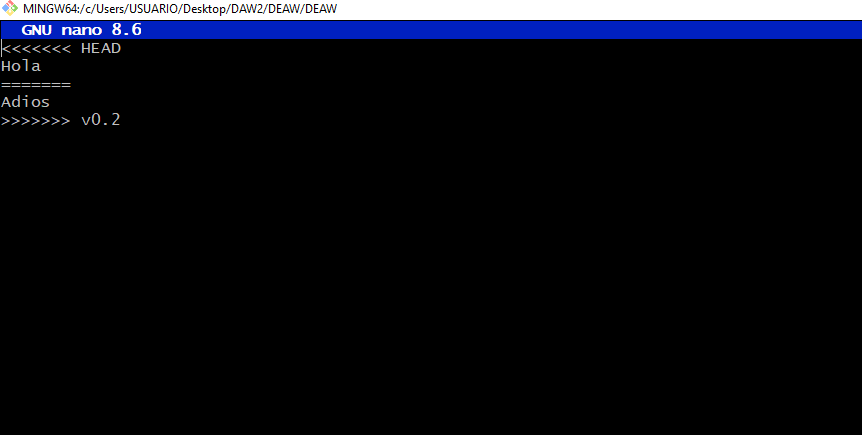




Aparece esto porque no añadí el fichero 1.txt

1. Posicionarse de nuevo en la rama **master** y hacer un merge con la rama **v0.2**

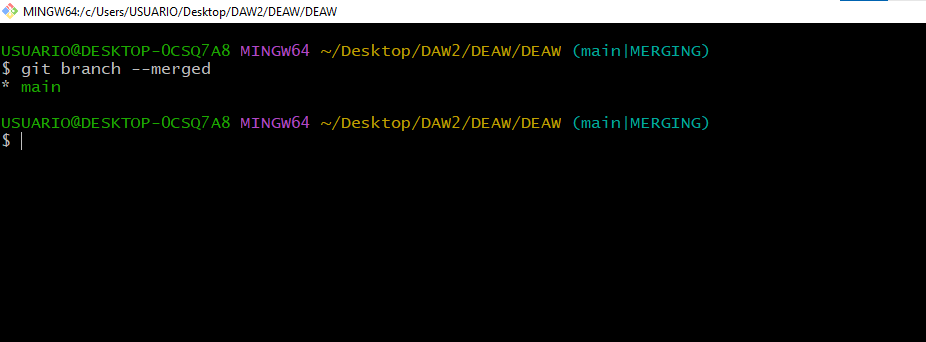




Comprobación del conflicto

Listado de ramas

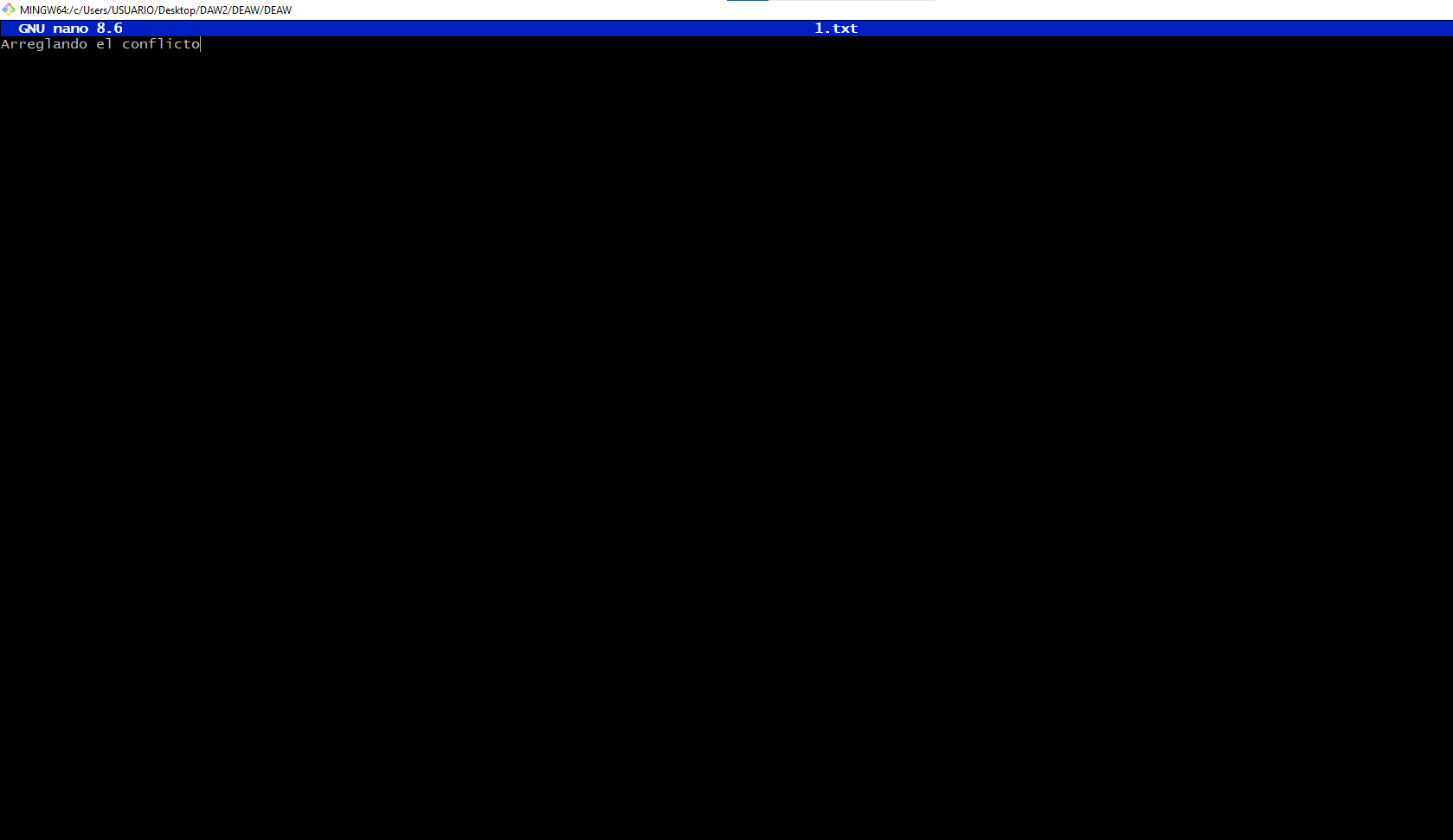
1. Listar las ramas con merge y las ramas sin merge.

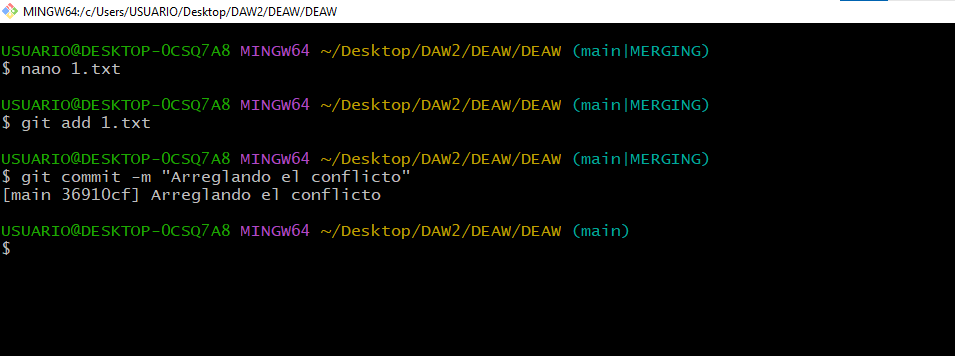




Arreglar conflicto

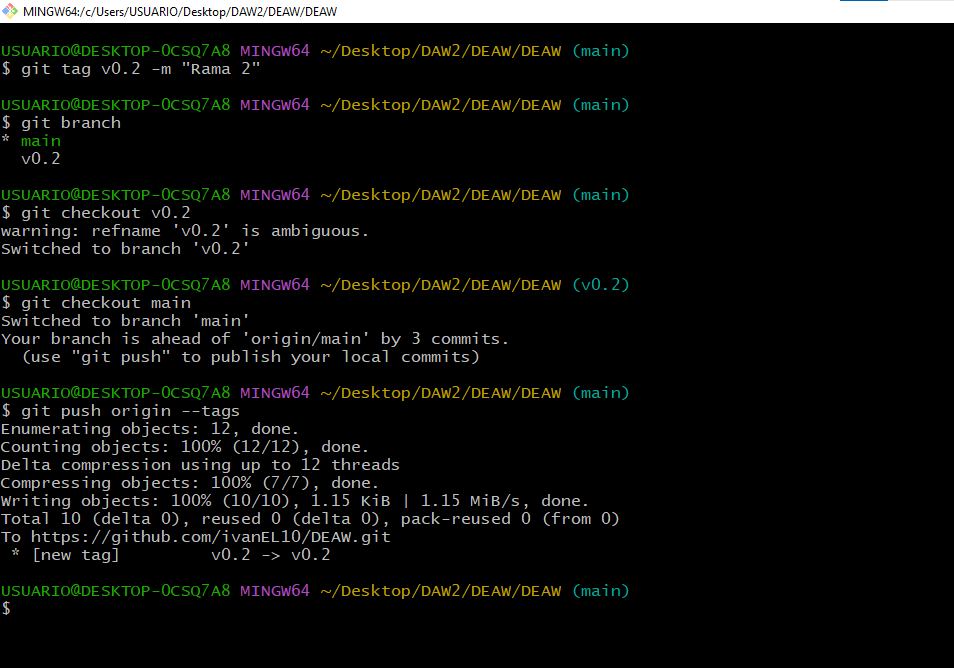
1. Arreglar el conflicto anterior y hacer un commit.



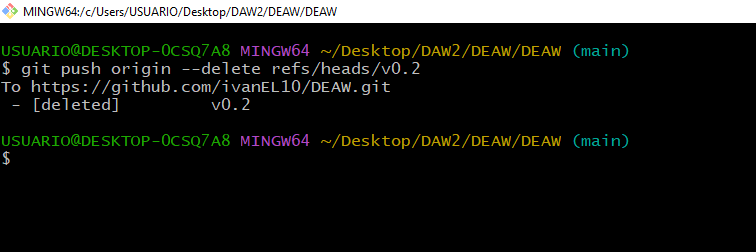


Borrar rama

1. Crear un tag **v0.2**



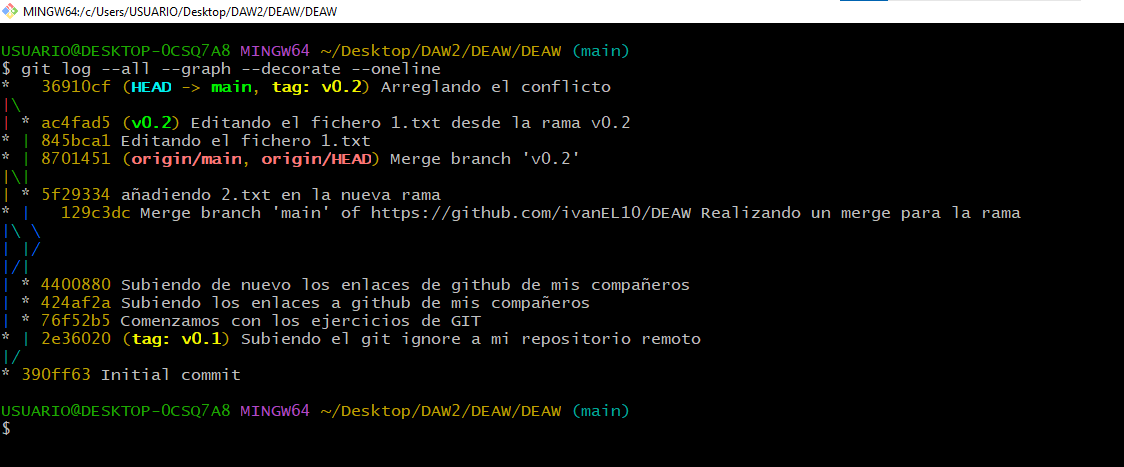
1. Borrar la rama **v0.2**



Tuve que poner este comando ya que el tag que le puse a la rama coincidía con el nombre y me detectaba 2 iguales y con este comando le especifico la rama que es

Listado de cambios

1. Listar los distintos commits con sus ramas y sus tags.



Referencias

[Pro Git book, written by Scott Chacon and Ben Straub and published by Apress](https://git-scm.com/book/es/v2)