PRACTICA GIT

1-Investigar 2comandos no vistos en clase (Ej: Git rebase, Git Fetch)

1.-Git revert

A veces, necesitaremos deshacer los cambios que hemos hecho. Hay varias maneras para deshacer nuestros cambios en local y/o en remoto (dependiendo de lo que necesitemos), pero necesitaremos utilizar cuidadosamente estos comandos para evitar borrados no deseados.

El comando git revert sirve para deshacer cambios efectuados en el historial de confirmaciones de un repositorio. Otros comandos para "deshacer" como, por ejemplo, git checkout y git reset, mueven los punteros de referencia HEAD y de la rama a una confirmación especificada. El comando git revert también toma una confirmación especificada, pero git revert no mueve los punteros de referencia a esta confirmación. Una operación de reversión tomará la confirmación especificada, invertirá los cambios de dicha confirmación y creará una "confirmación de reversión" nueva. Entonces, los punteros de referencia se actualizarán para apuntar a la nueva confirmación de reversión, lo cual la convertirá en la punta de la rama.

2.- Git merge

La fusión es la forma que tiene Git de volver a unir un historial bifurcado. El comando git merge permite tomar las líneas independientes de desarrollo creadas por git branch e integrarlas en una sola rama.

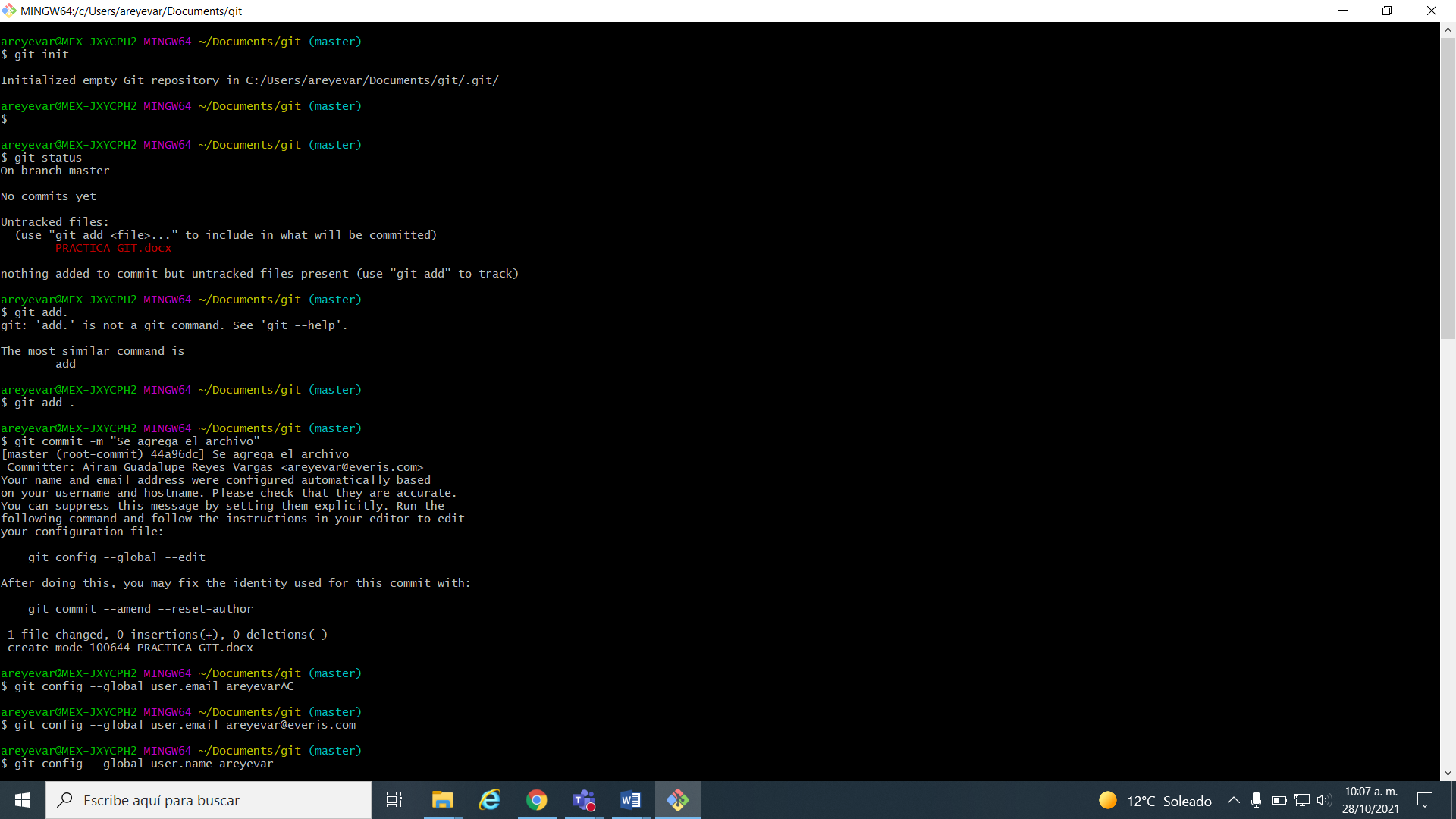
Git merge combinará varias secuencias de confirmaciones en un historial unificado. En los casos de uso más frecuentes, git merge se utiliza para combinar dos ramas. Git merge toma dos punteros de confirmación, normalmente los extremos de la rama, y encuentra una confirmación base común entre ellos. Una vez que Git encuentra una confirmación base en común, crea una "confirmación de fusión" nueva que combina los cambios de cada secuencia de confirmación de fusión puesta en cola.

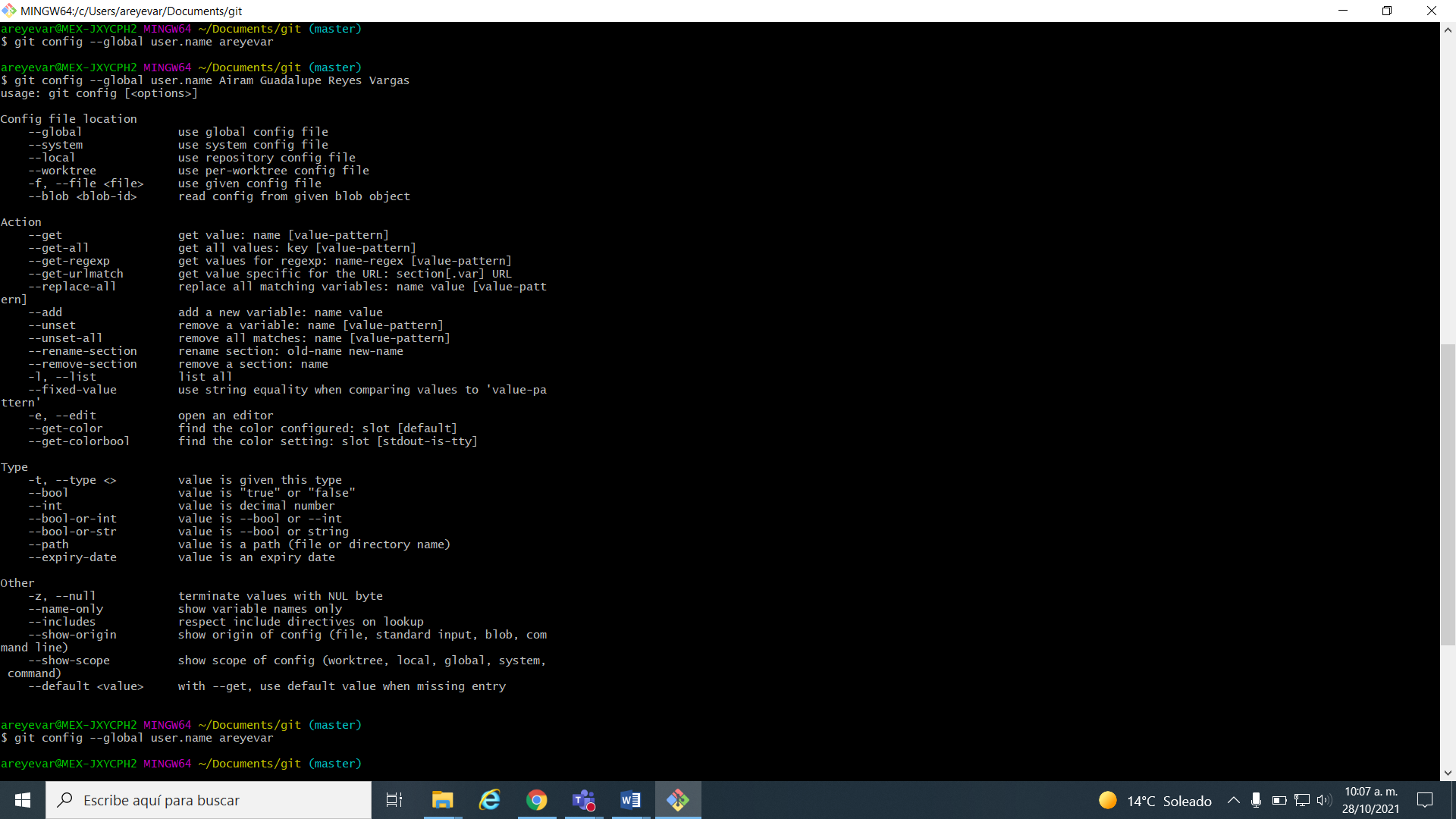
2.- Investigar GitFork y cómo funciona

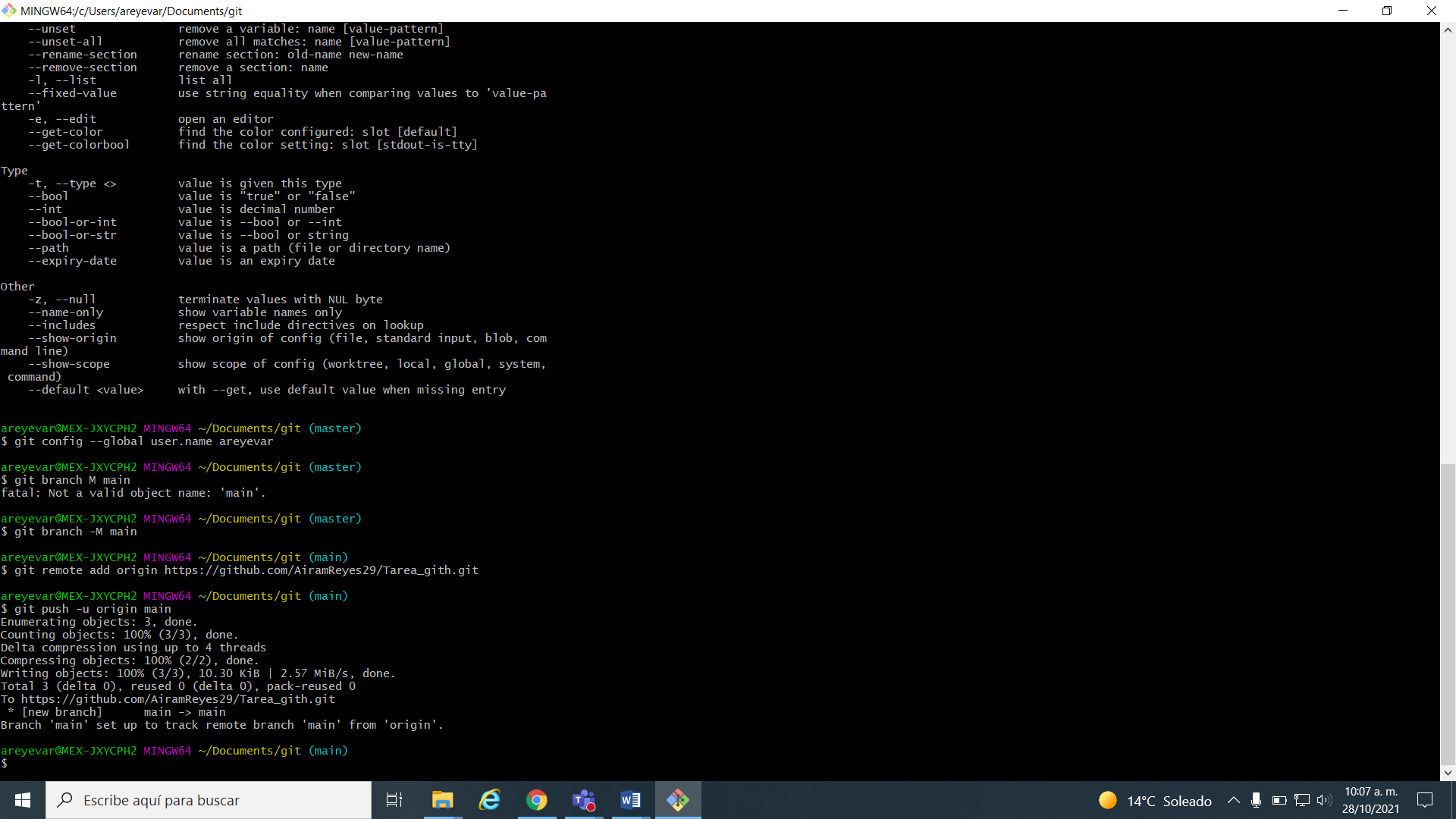
El "fork" es una de las operativas comunes con el trabajo en Git y GitHub. Básicamente sirve para crear una copia de un repositorio en tu cuenta de usuario. Ese repositorio copiado será básicamente un clon del repositorio desde el que se hace el fork, pero a partir de entonces el fork vivirá en un espacio diferente y podrá evolucionar de manera distinta, a tu propio cargo.

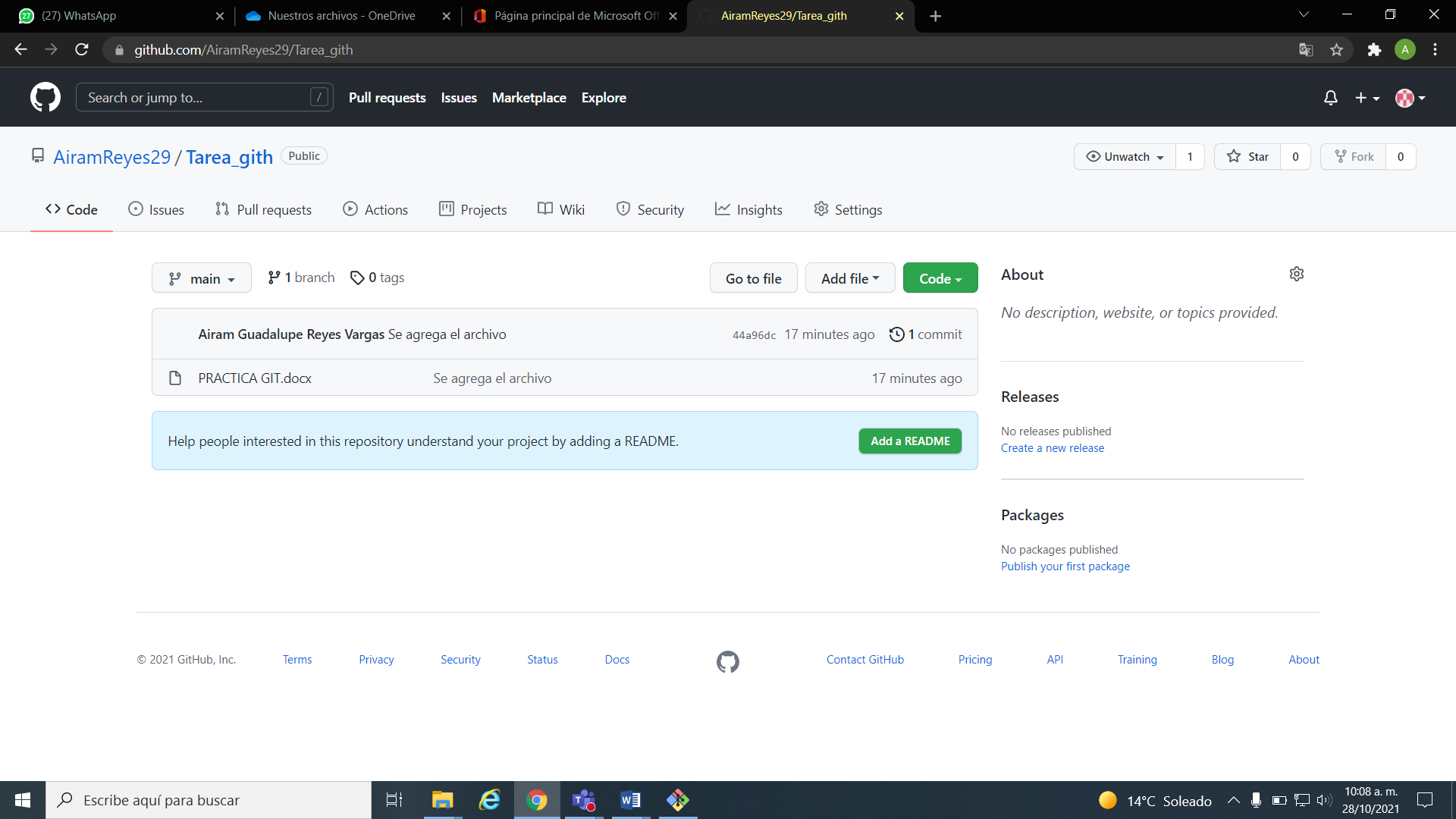
La palabra fork se traduce al castellano, dentro del contexto que nos ocupa, como bifurcación. Cuando hacemos un fork de un repositorio, se hace una copia exacta en crudo (en inglés «bare») del repositorio original que podemos utilizar como un repositorio git cualquiera.

Cuando haces un fork de un repositorio, se crea un nuevo repositorio en tu cuenta de Github o Bitbucket, con una URL diferente (fork). Acto seguido tienes que hacer un clon de esa copia sobre la que empiezas a trabajar de forma que cuando haces push, estás modificando TU COPIA (fork). El repositorio original sigue intacto.









Me agrado tu tarea hasta con imágenes y todo, agregamos contenido para la rama feature1 y esperar hacer un merge con develop.