GIT



Contornos de Desarrollo

Nicolas Primoy Pucheta

Ejercicio 2.1 Instalación y primeros pasos en Git

Primeros pasos

Una vez instalado Git, realizamos la configuración utilizando los siguientes comandos:

*git config --global user.name "Tu Nombre"*

*git config --global user.email "*[*tu.email@example.com*](mailto:tu.email@example.com)*"*

*git config --global core.editor "visual code studio”*

Verificamos que la configuración sea haya aplicado correctamente con el comando :

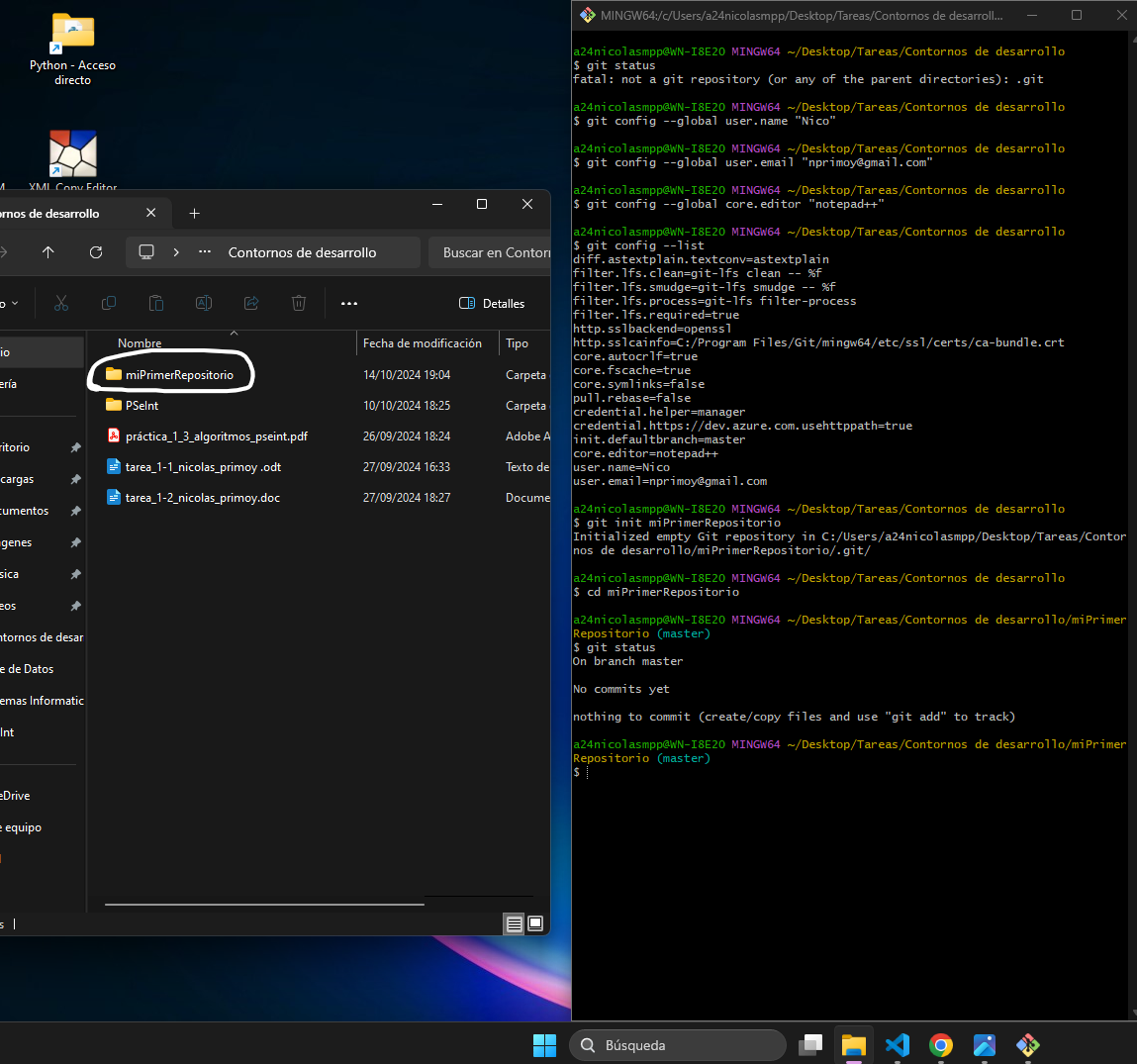
*git config –list*

Creamos nuestro primer repositorio con el comando:

*git init miPrimerRepositorio*

Comprobamos el estado del repositorio utilizando el comando:

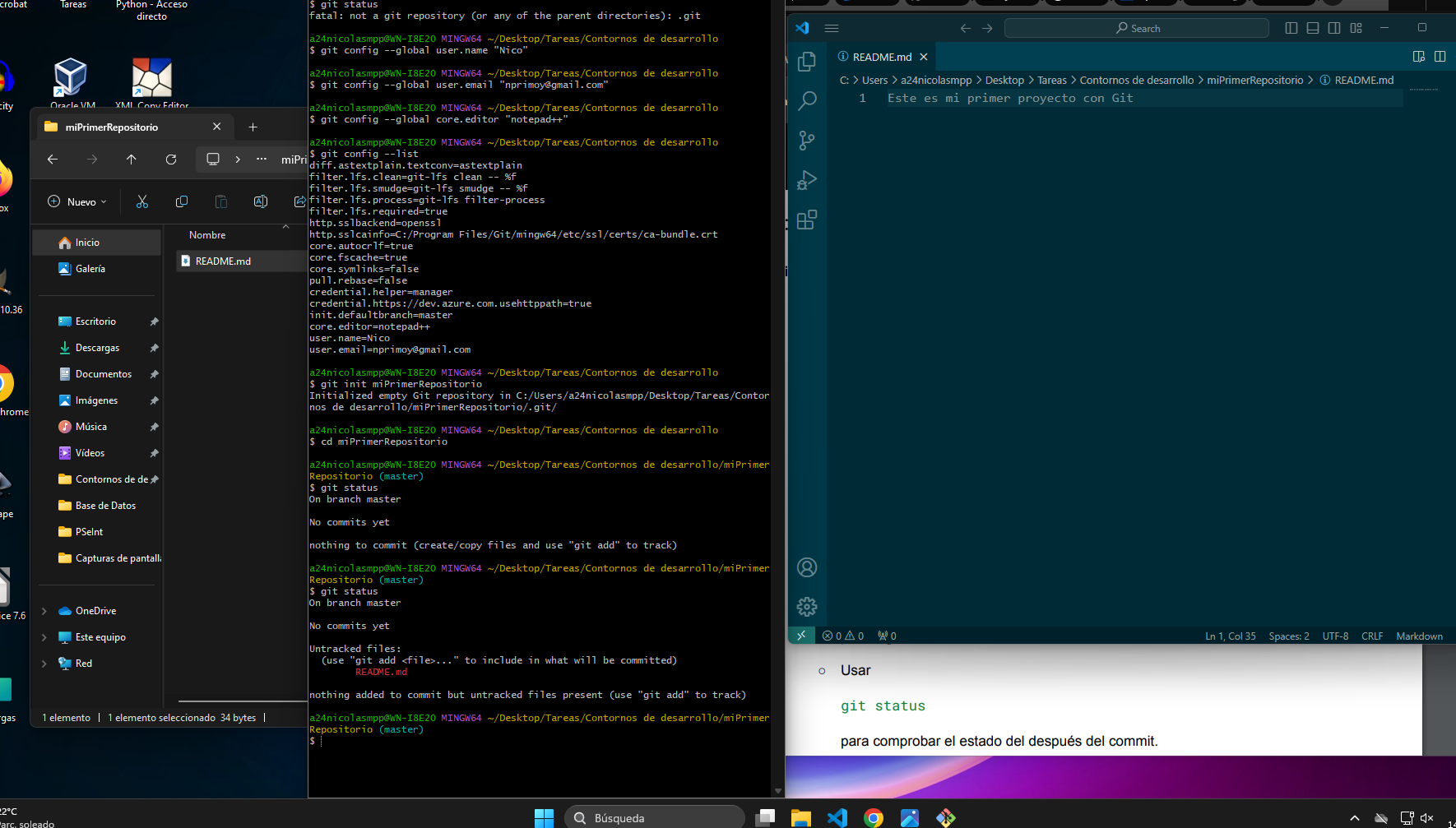
*git status*

**

Primer commit

Para realizar nuestro primer commit creamos un archivo “README.md” en nuestra carpeta de git, con una descripción.

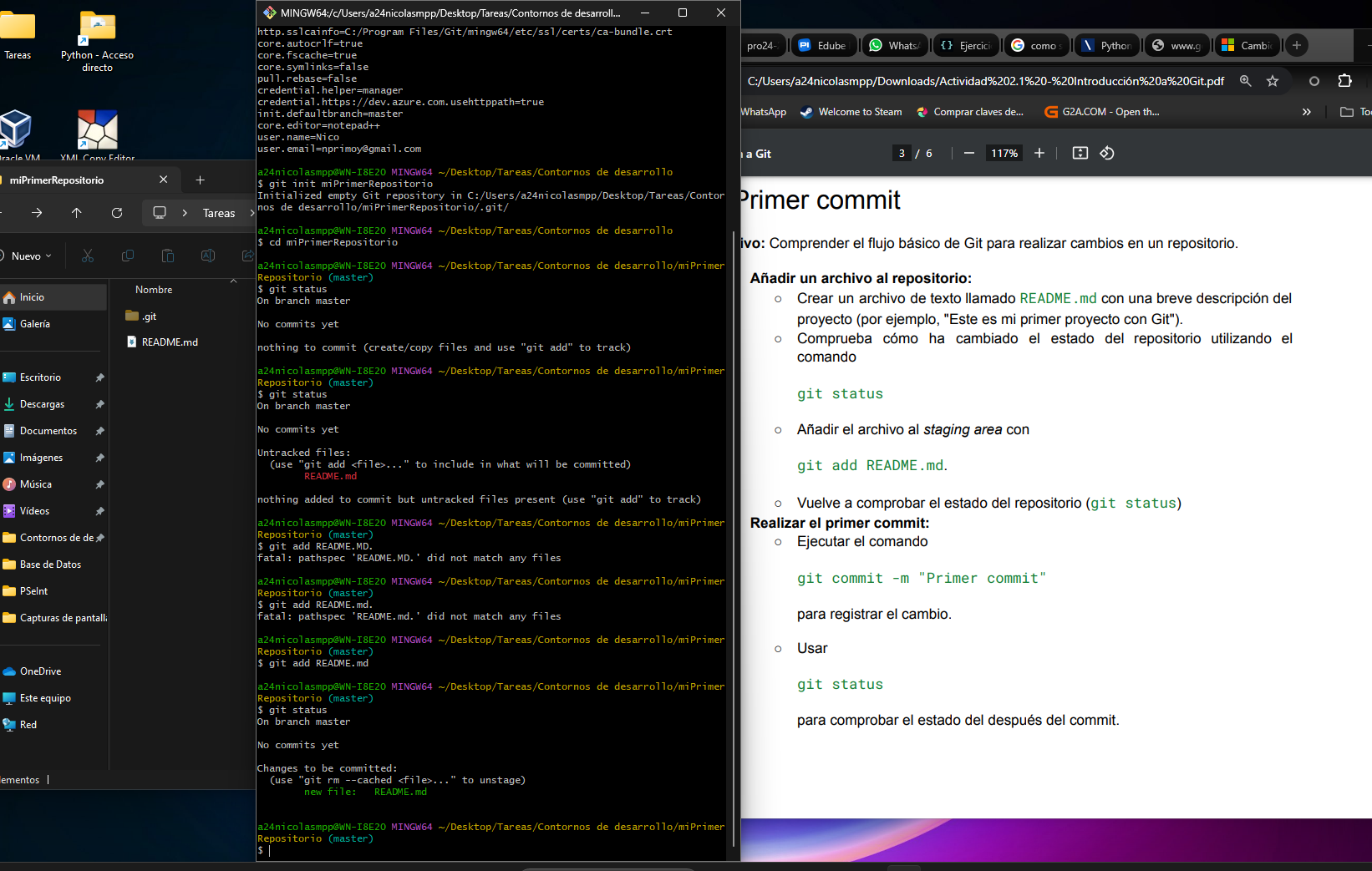
Verificamos el estado del repositorio utilizando: *git status*



Para añadir el archivo al staging area utilizamos el siguiente comando:

*git add README.md*

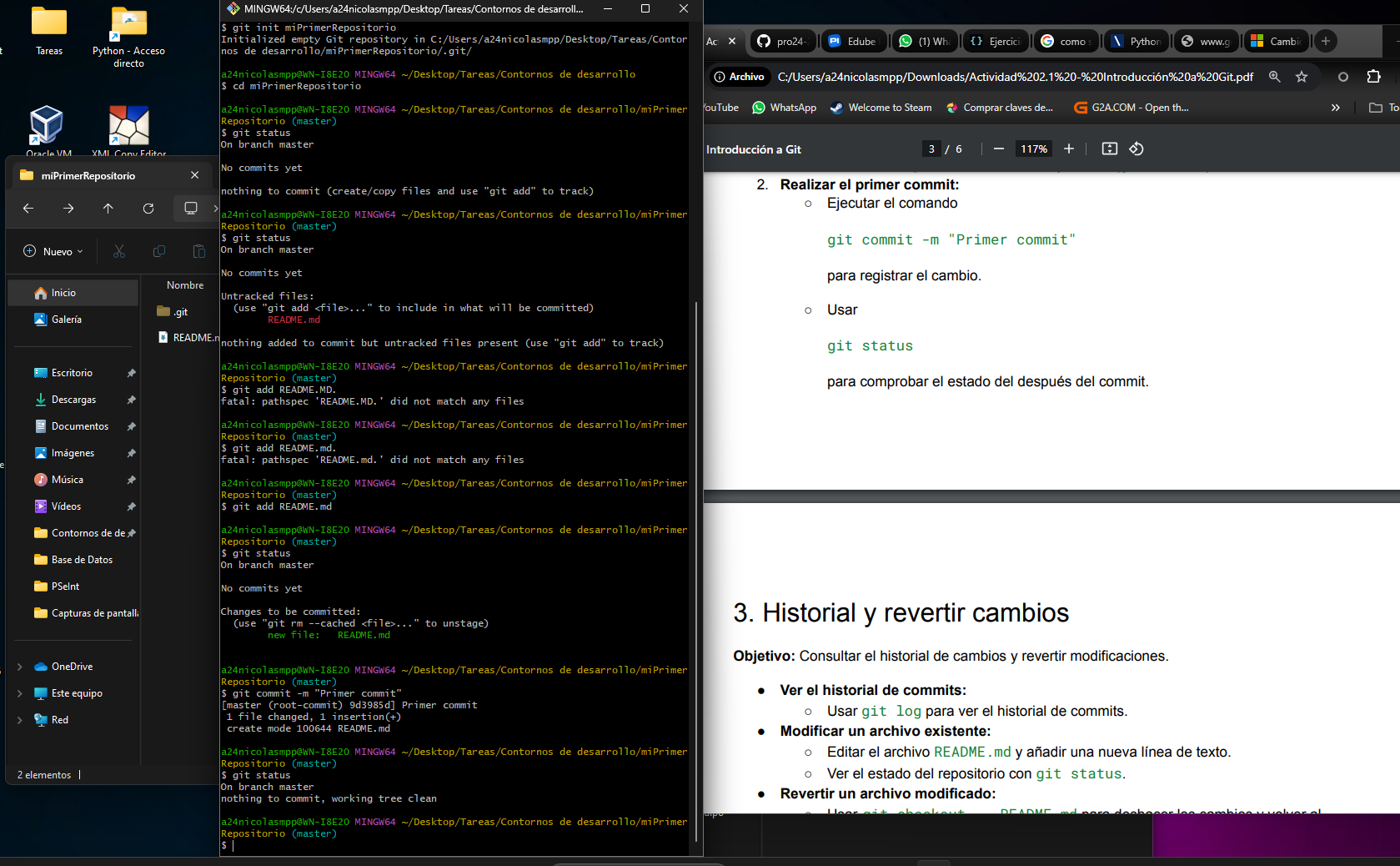
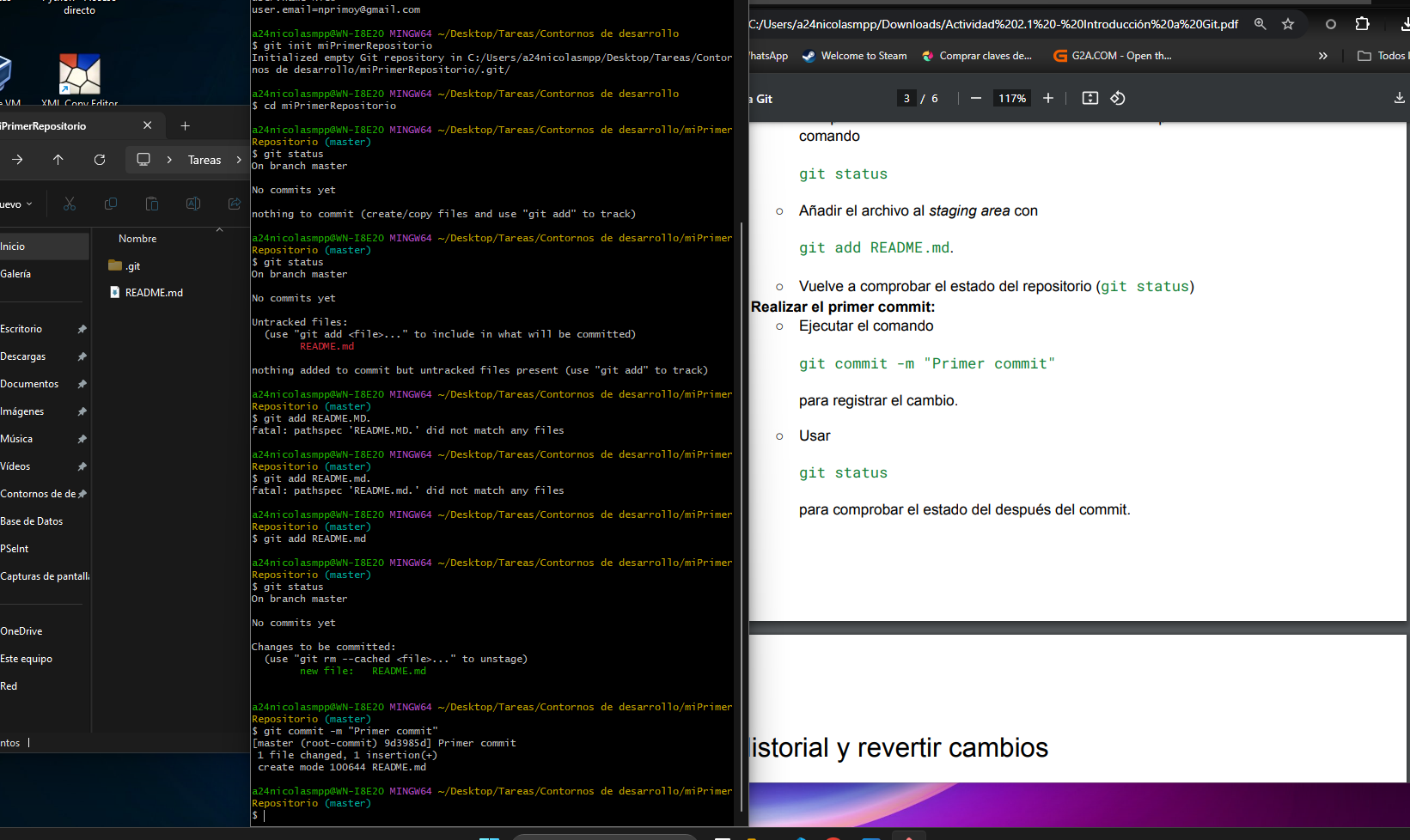
Y volvemos a verificar el estado con: *git status*



Realizamos nuestro primer commit y verificamos los cambios:

*git commit -m "Primer commit"*

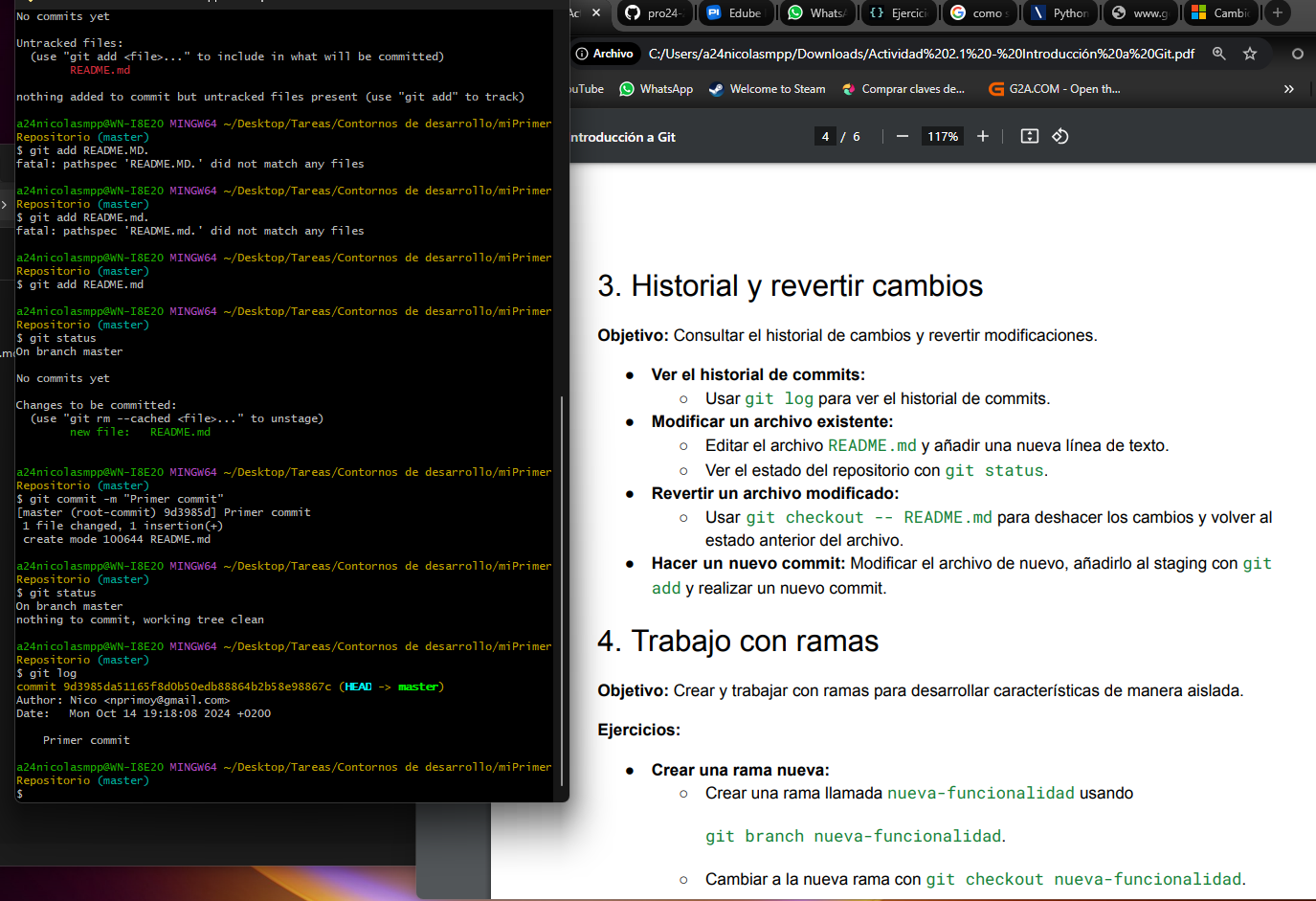
*git status*



Historial y revertir cambios

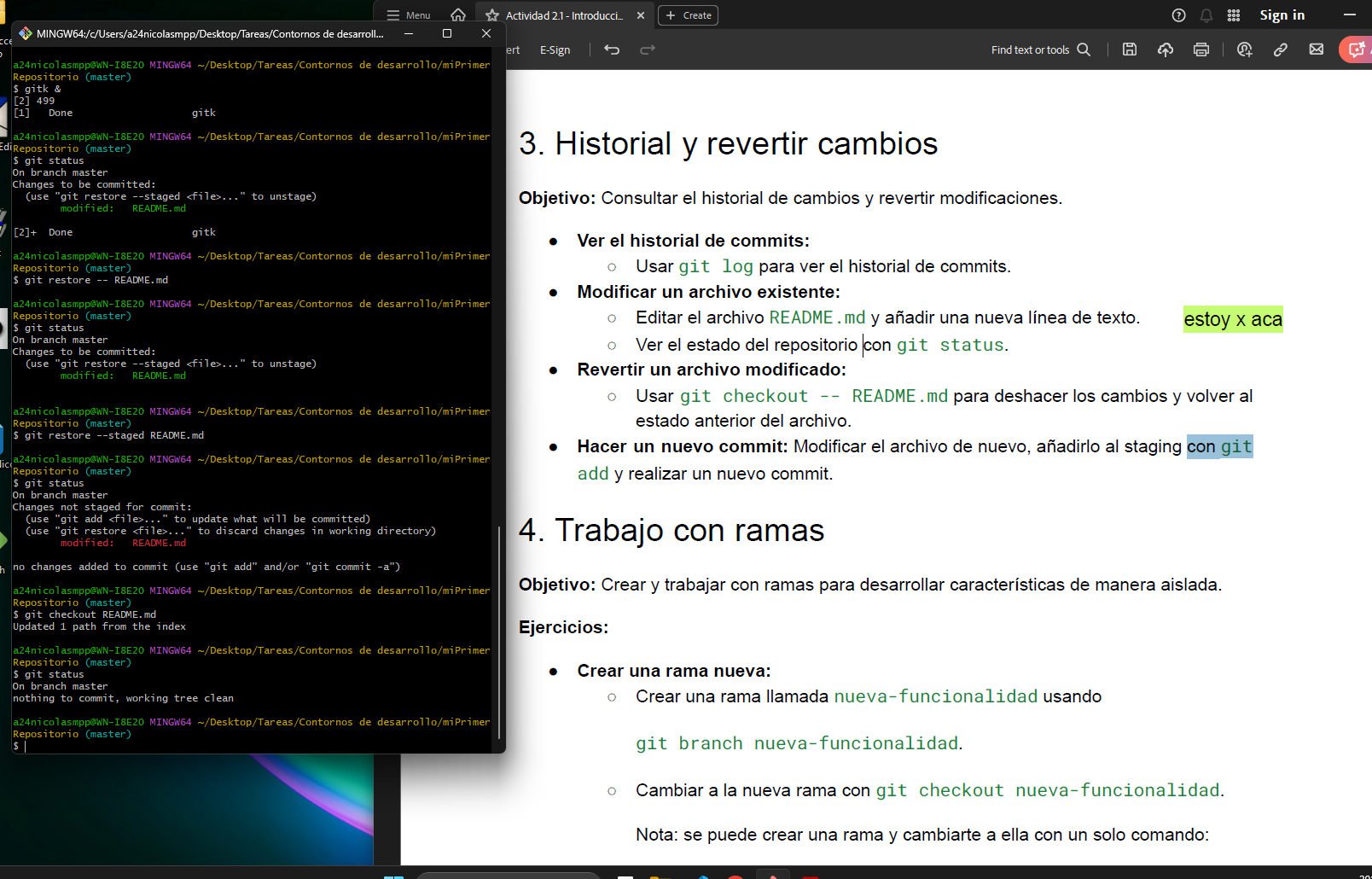
Comprobamos el historial de commits con el siguiente comando:

*git log*

**

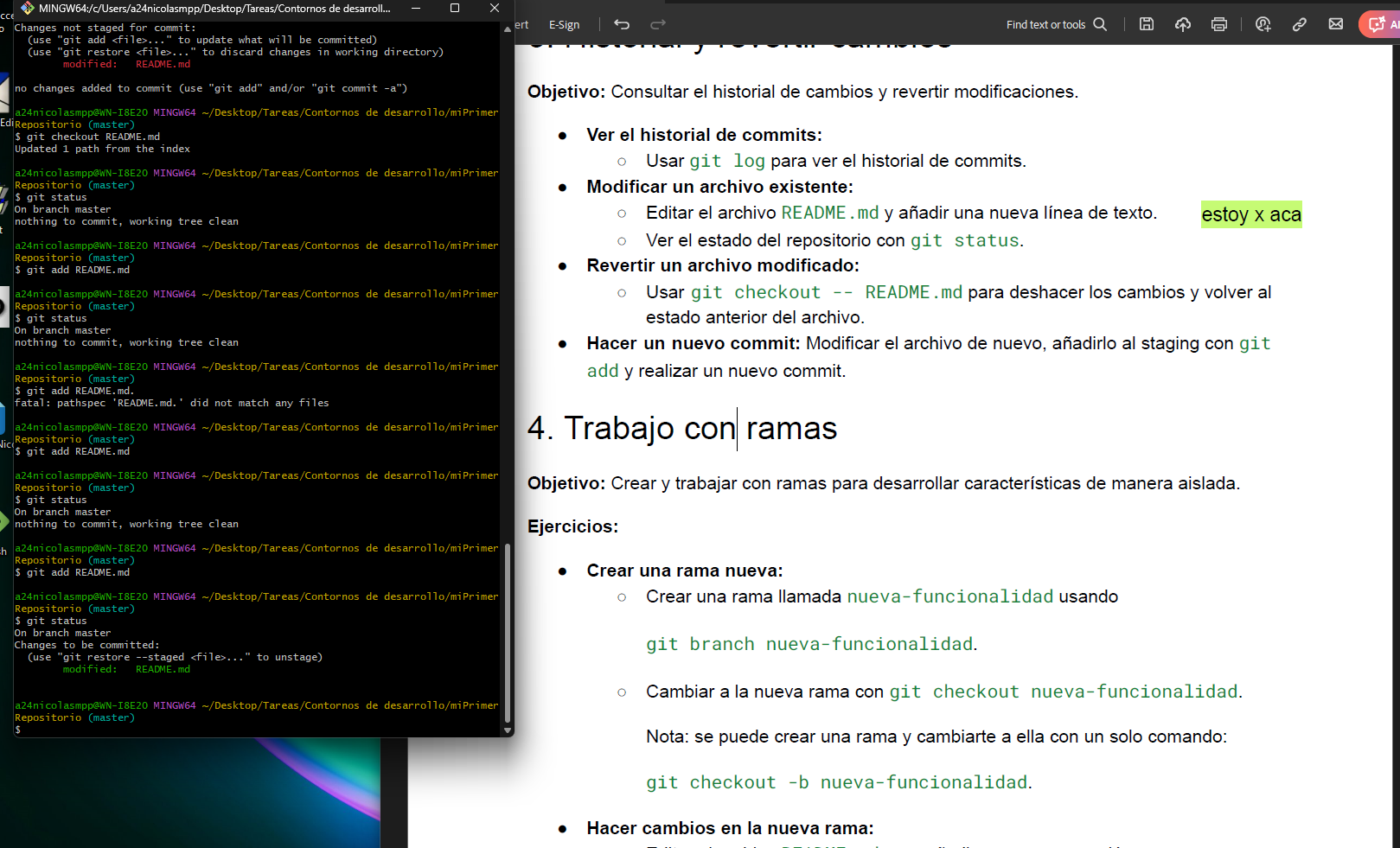
Realizamos una modificación en el archivo y verificamos el estado con *git status*

En la imagen se puede comprobar que el estado del README.md se ve en rojo ya que esta modificado en local pero no actualizado.

**

Agregamos la modificación con el comando  *git add README.md*

Al estar actualizado vuelve a estar en verde

**

Trabajo con ramas

Creamos una rama llamada “nueva-funcionalidad”, cambiamos a la rama y verificamos el estado usando los siguientes comandos:

*git branch nueva-funcionalidad*

*git checkout nueva-funcionalidad*

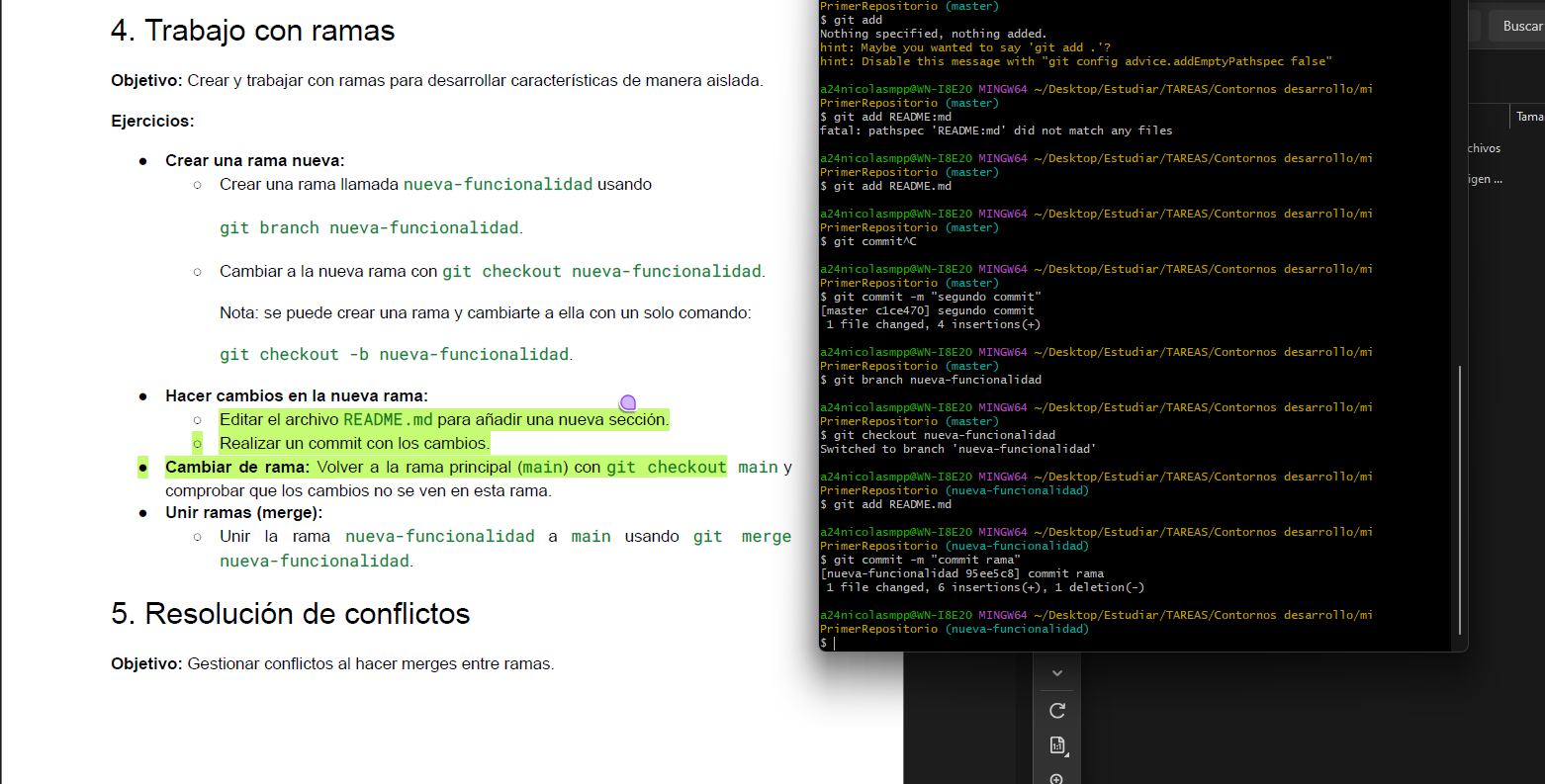
*git status*

Realizamos cambios en el archivo README.md, lo agregamos, subimos con un commit y verificamos estado:

*git add README.md*

*git commit -m "commit rama"*

*git status*

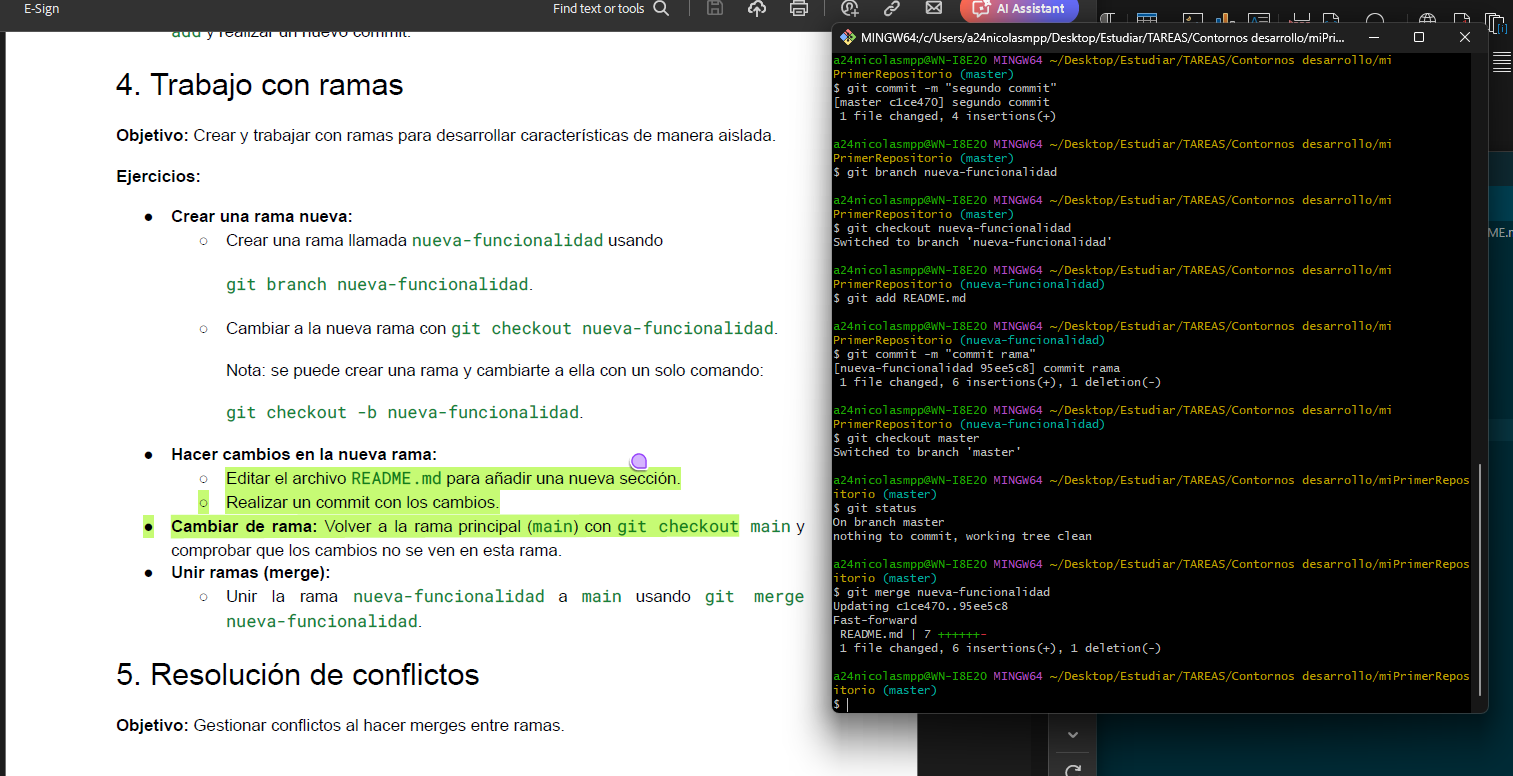


Volvemos al master, verificamos el estado y realizamos un merge de la rama “nueva-funcionalidad”:

*git checkout master*

*git status*

*git merge nueva-funcionalidad*

**

Buscando el error realizamos un commit en el master volvemos a cambiar a la rama y realizamos otro commit en la misma linea que el commit en el master:

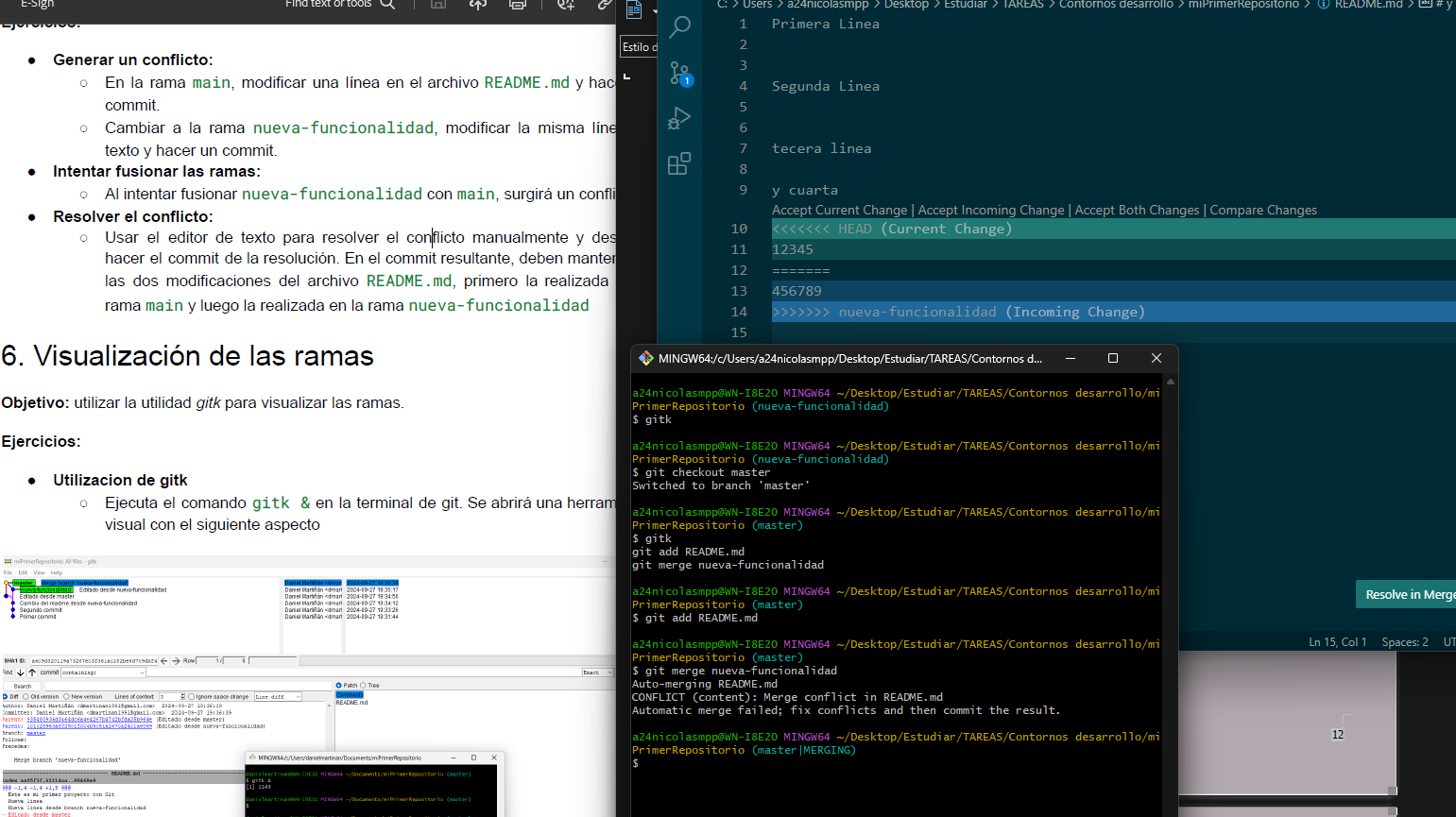
*git add README.md*

*git commit -m “tercer commit master”*

*git checkout nueva-funcionalidad*

*git add README.md*

*git commit -m “commit rama2”*

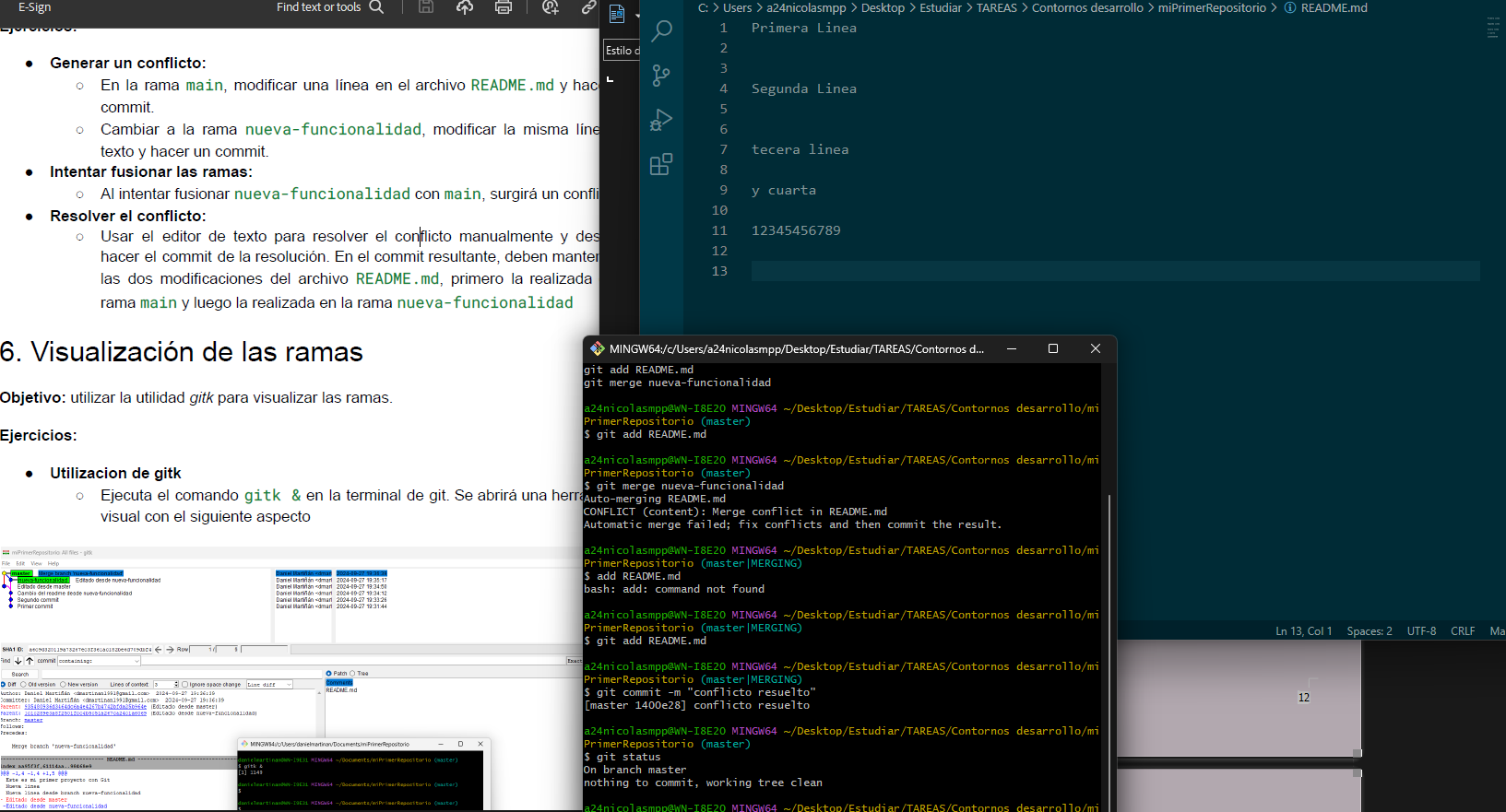
**

Resolvemos el conflicto desde el editor de texto y volvemos a subir el archivo:

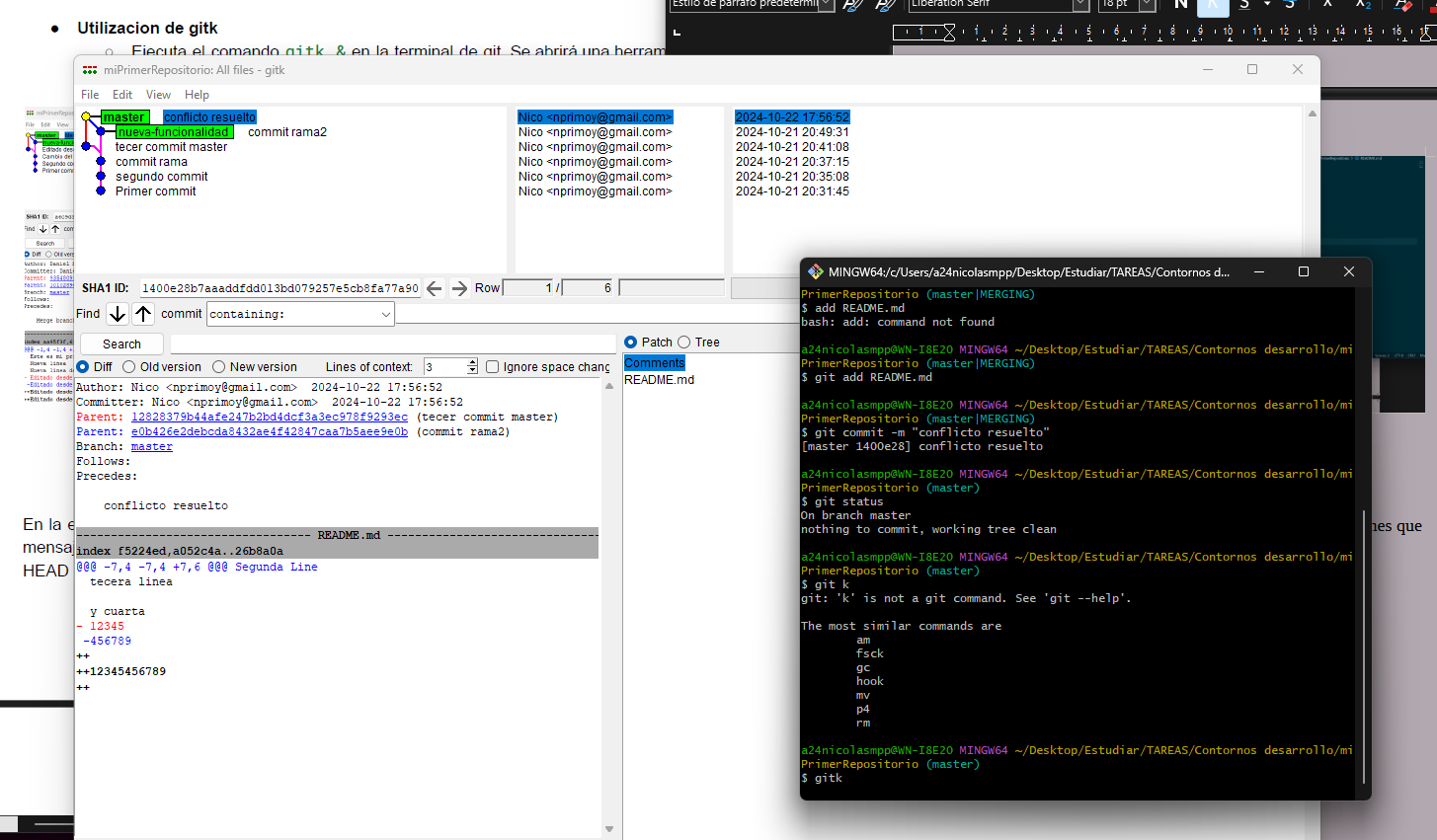
*git add README.md*

*git commit -m “conflicto resuelto”*

*git status*

**

Utilizamos el comando *gitk* para visualizar las modificaciones que se realizaron en la reposición.

**