

GUIÓN DE LA ACTIVIDAD AEV3:

Título

Git - Ramas y repositorios remotos

Objetivos

- Utilizar ramas en Git para el desarrollo de un proyecto.
- Utilizar flujos de trabajo con ramas y un repositorio local.
- Detectar y resolver conflictos en la integración de código.
- Crear repositorios remotos mediante Git.
- Conocer flujos de trabajo sincronizando repositorios remotos y locales.

Temporalización

Se estima una dedicación de **5 horas**. Teniendo en cuenta que habrá que revisar los recursos facilitados en el curso en Florida Oberta para poder de realizar la actividad.

Proceso de desarrollo

1. Se van a proponer una serie de pasos en los que habrá que realizar una tarea específica.
2. Conforme se realice cada paso, se realizará una captura de pantalla con las explicaciones oportunas en un documento, de modo que se argumente o confirme que el paso ha sido realizado.
3. Entregar un documento PDF, debidamente identificado, que incluya cada enunciado de cada paso con la respuesta correspondiente, a través de Florida Oberta.

Evaluación

La actividad se ha subdividido en pasos. Cada uno de ellos dispondrá de una valoración en puntos, en función de su dificultad o esfuerzo requerido. En total sumarán 10 puntos. Como norma, cada paso se valorará del siguiente modo:

- Errores graves: la respuesta no corresponde a lo solicitado. El paso puede llegar a sumar 0 puntos.
- Errores leves:
 - Paso parcialmente incorrecto, resta entre 0.25 - 1 punto.
 - Cada fallo leve debido a un motivo menor, resta entre 0.25 - 0.50 puntos.

Recursos

Puestos a disposición del alumno en el curso correspondiente del campus virtual Florida Oberta.

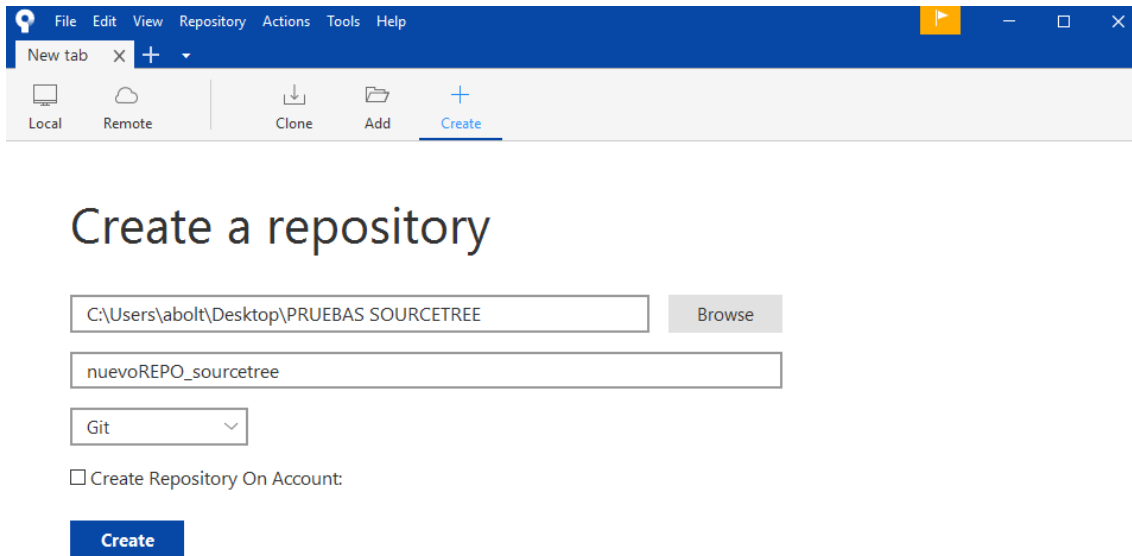
Detalle de la actividad

Supón que formas parte de un equipo de desarrollo de software y, entre tus funciones, está la de administrar el repositorio que se usa para gestionar la documentación de un proyecto, por ejemplo, la modernización de un portal web para la gestión de pedidos de clientes de una compañía de venta online:

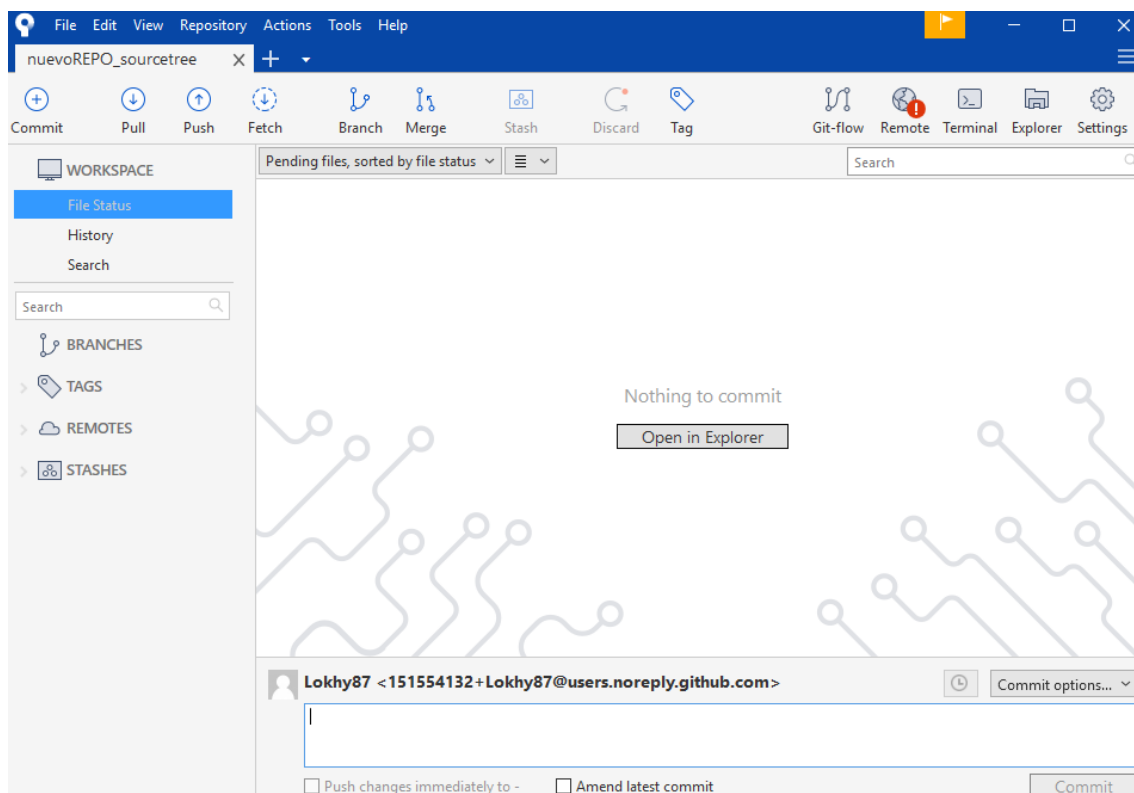
1. Desarrollo de la nueva pantalla de pedidos de cliente (3 puntos)

- Crea un nuevo directorio en tu sistema de ficheros local.

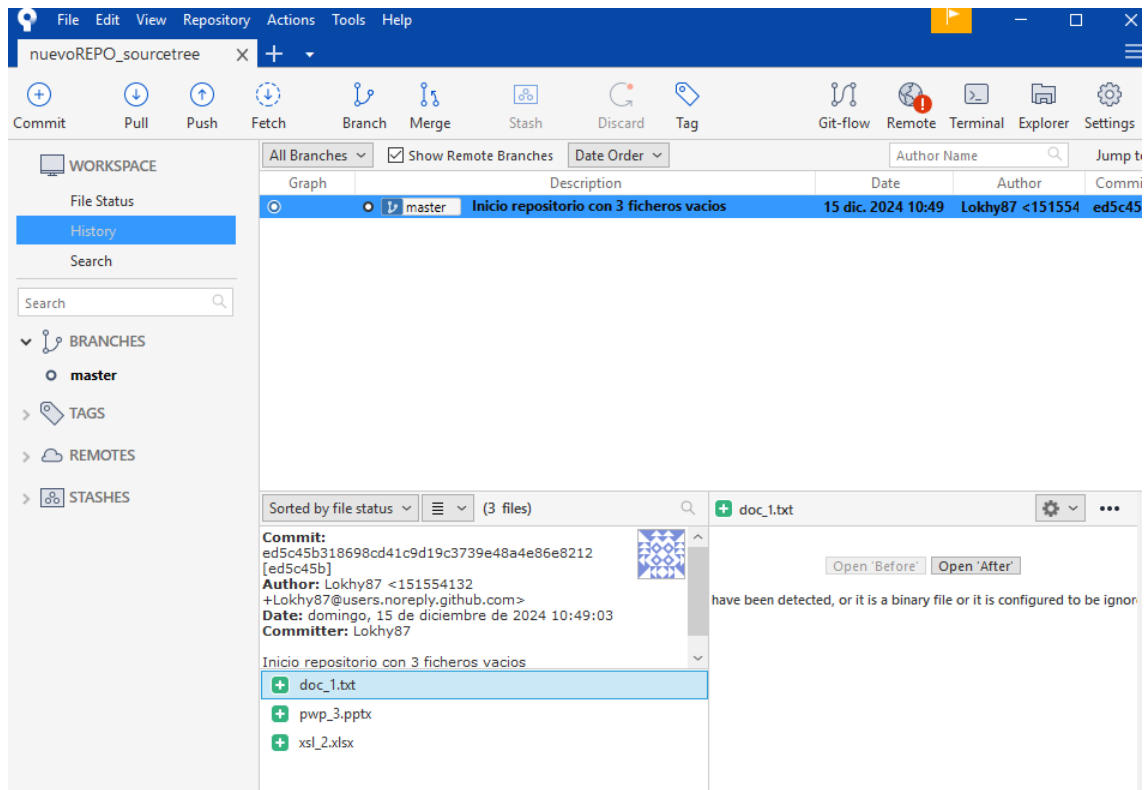
He creado un repositorio mediante la opcion "Create" añadiendo la ruta de acceso y el nombre, "nuevoREPO_sourcetree"



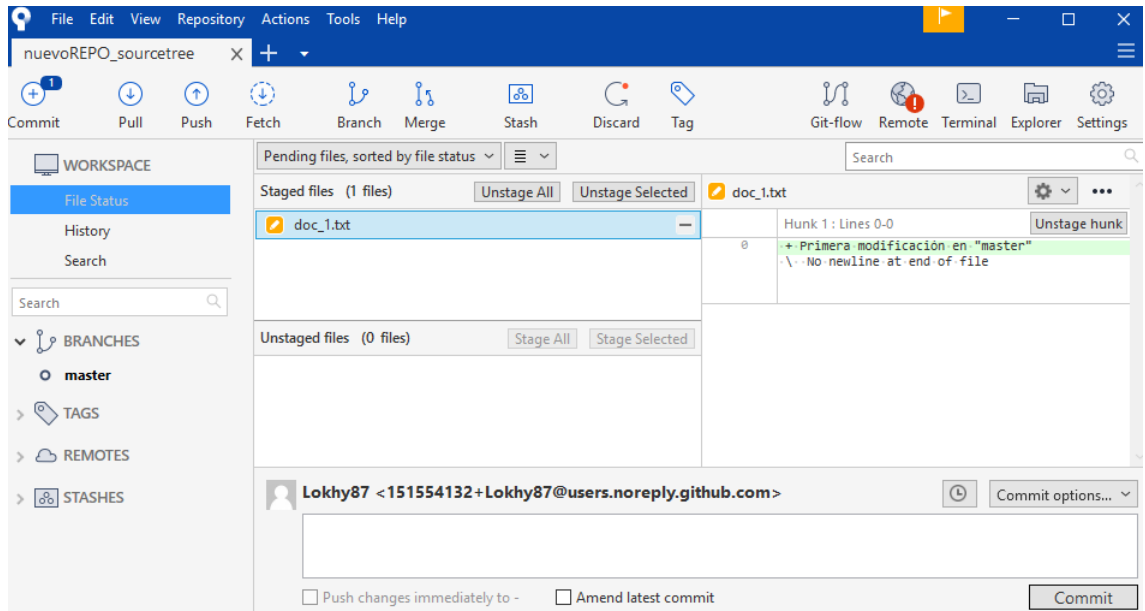
- Inicializa un repositorio en él.



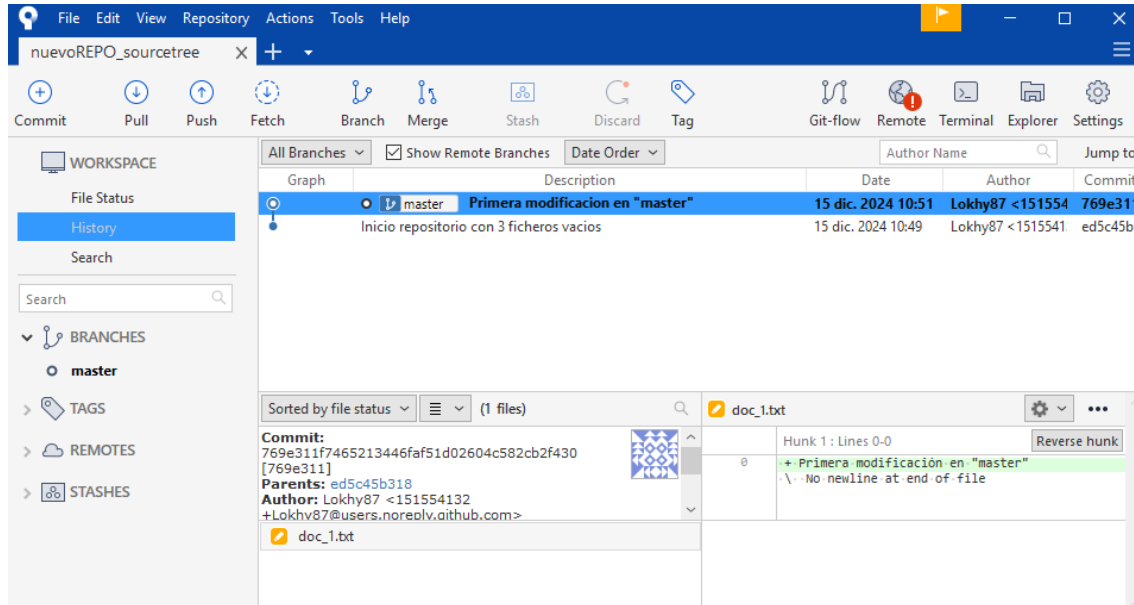
- Añade un mínimo de 3 ficheros vacíos al repositorio.



- Haz modificaciones en los ficheros y actualiza el repositorio mediante una consolidación.

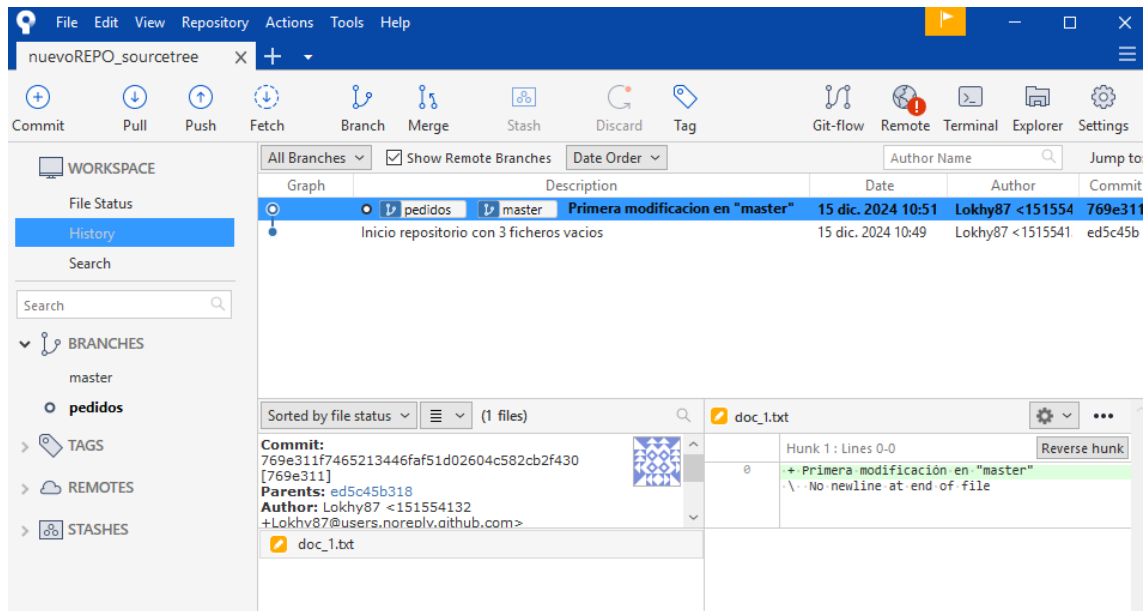


En la imagen se observa la modificacion del documento. A continuacion hago lo paso a parte de preparacion y hago la consolidacion con un commit.



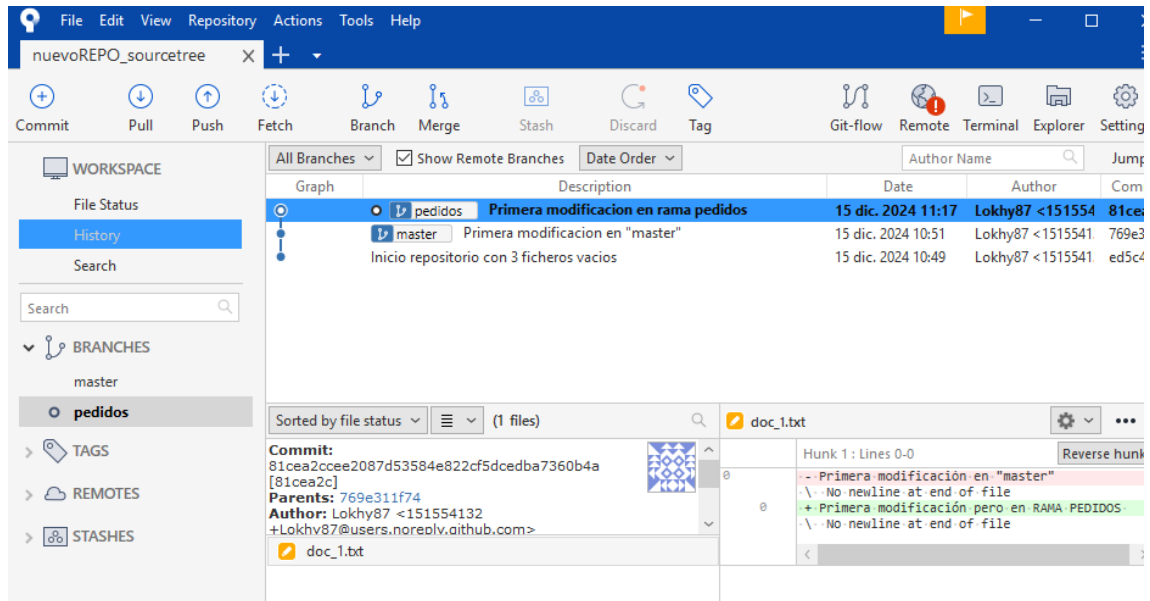
- El equipo va a comenzar a desarrollar la nueva pantalla de pedidos y va a requerir una línea de trabajo específica. Crea una rama para esta funcionalidad, por ejemplo, llámala “pedidos”.

Posicionandome en “master”, creo una rama



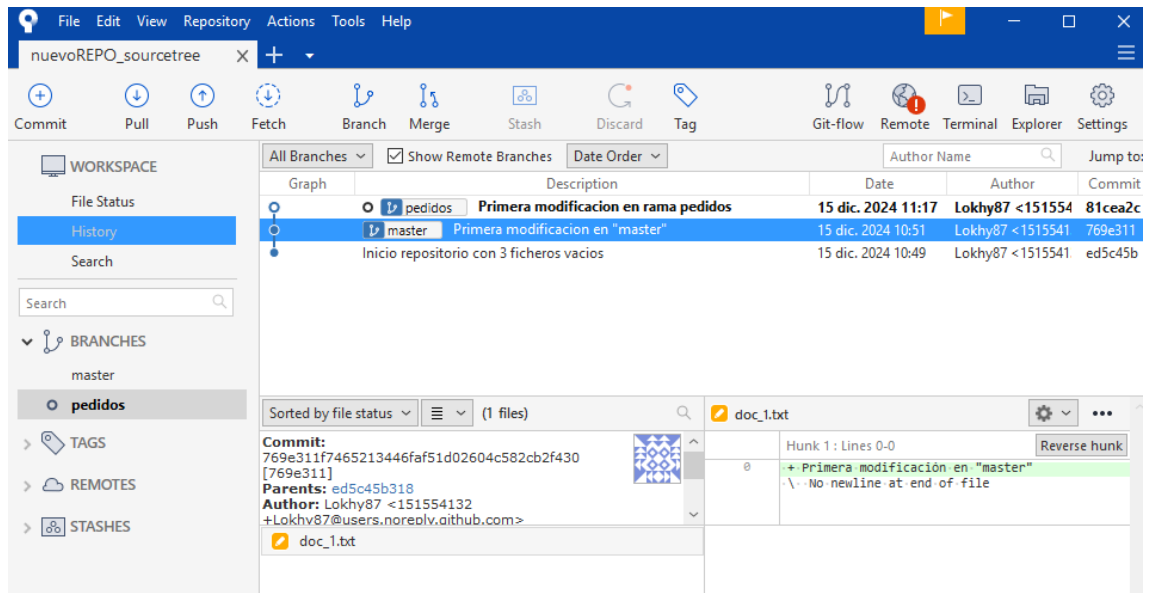
- Con la rama “pedidos” activa, realiza un cambio en algún fichero y consolida los cambios en dicha rama.

Primera modificacion



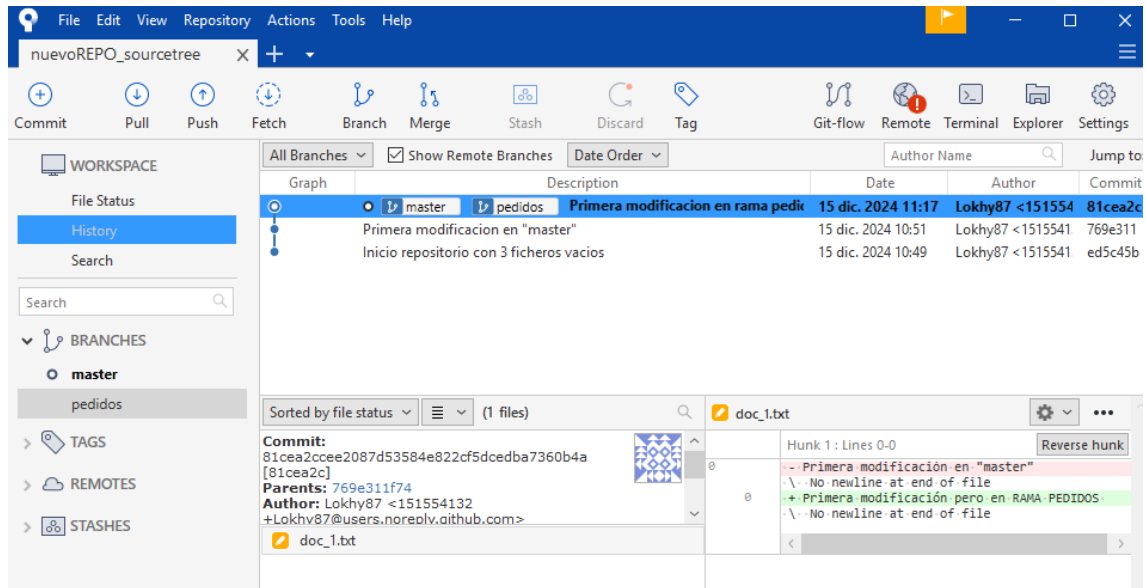
- Suponemos que hemos terminado de desarrollar la nueva pantalla de pedidos. Comprueba que los cambios realizados en la rama “pedidos”, no existen en los ficheros de la rama “master”.

Aquí podemos comprobar que el cambio en “pedidos” no afecta a “master”



- Integra los cambios de la rama “pedidos” en la rama “master”.

Posicionándome en “master”, hago el merge en rama “pedidos”.

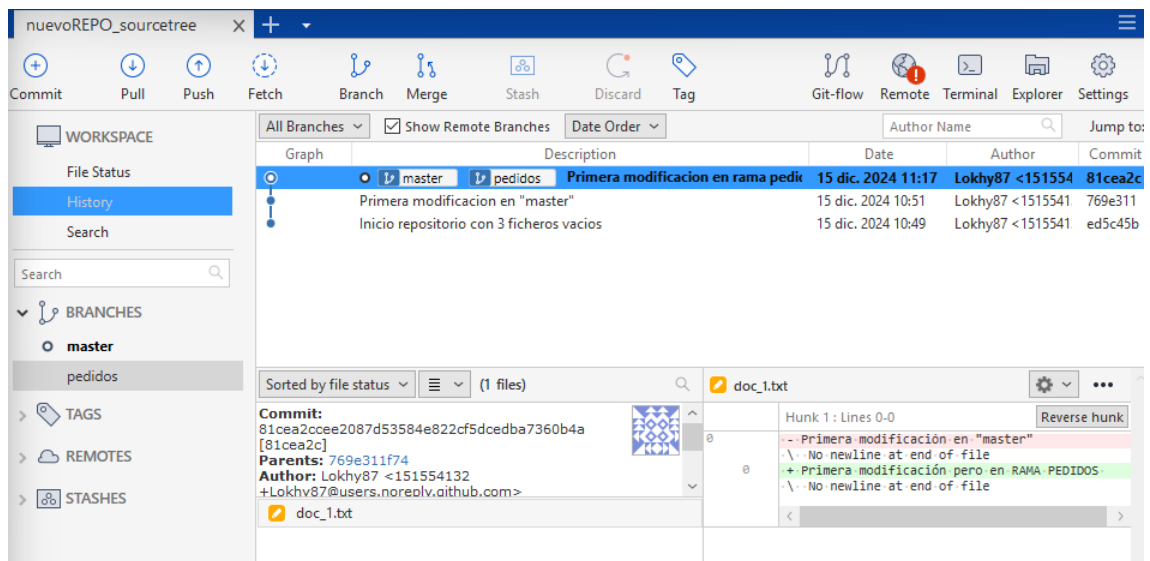


- Explica qué tipo de fusión has llevado a cabo.

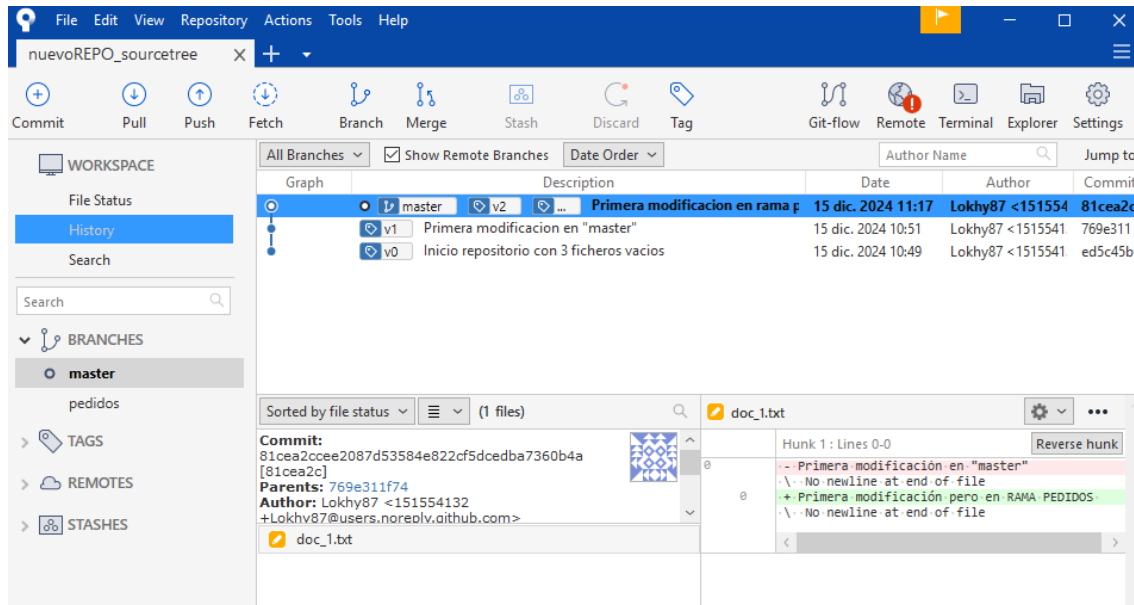
Lo que tenía en la rama “pedidos” se ha volcado en “master” para continuar con el proyecto, integrando los cambios realizados en otra rama. Es un tipo de fusión **merge fast forward**.

- Después de integrar, comprueba si ahora existen en la rama “master”, los cambios que se habían realizado en la rama “pedidos”.

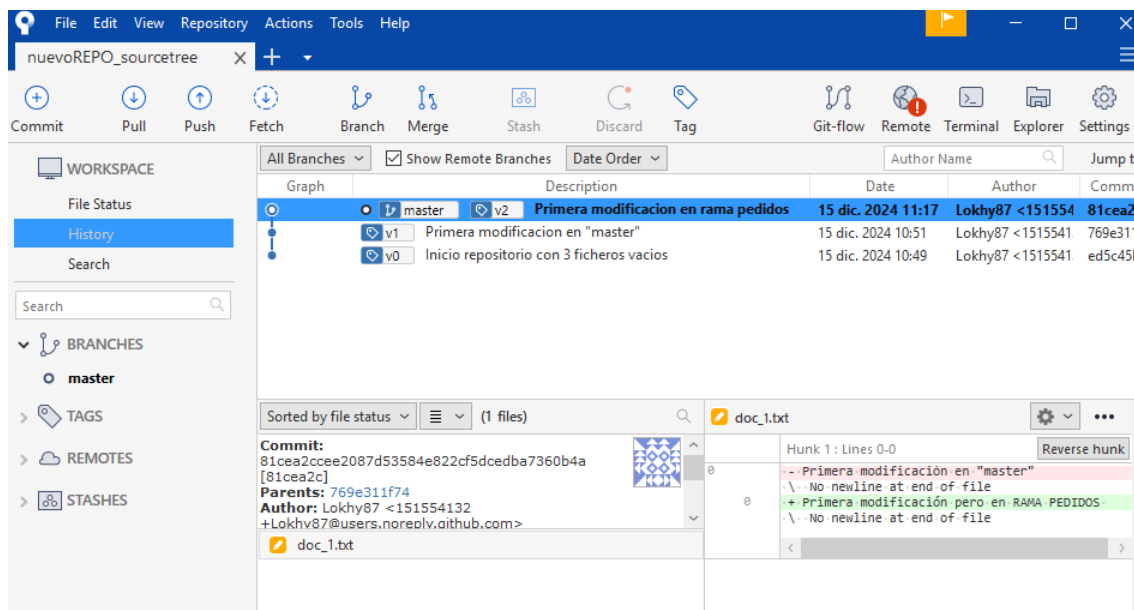
Aquí se comprueba que la modificación de “pedidos” se ha integrado en “master”.



- Asigna una etiqueta a cada consolidación con un número de versión (v0 la más antigua, v1 la posterior, v2 la más reciente).

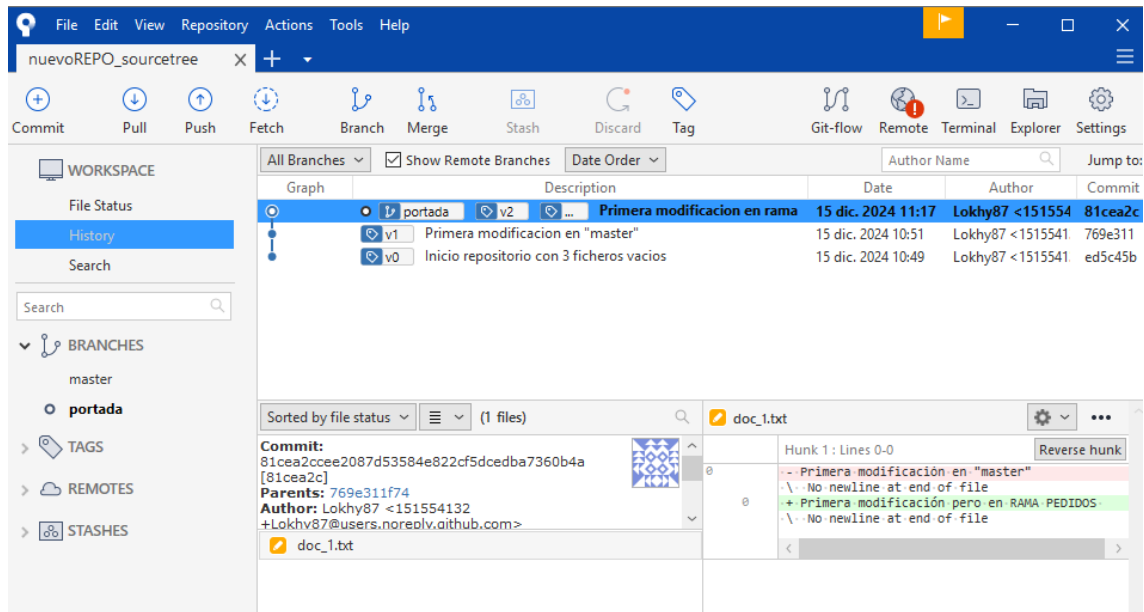


- Llegados a este punto, como hemos terminado la nueva pantalla de pedidos y la hemos integrado en la rama "master", podemos borrar la rama "pedidos", elimínala.

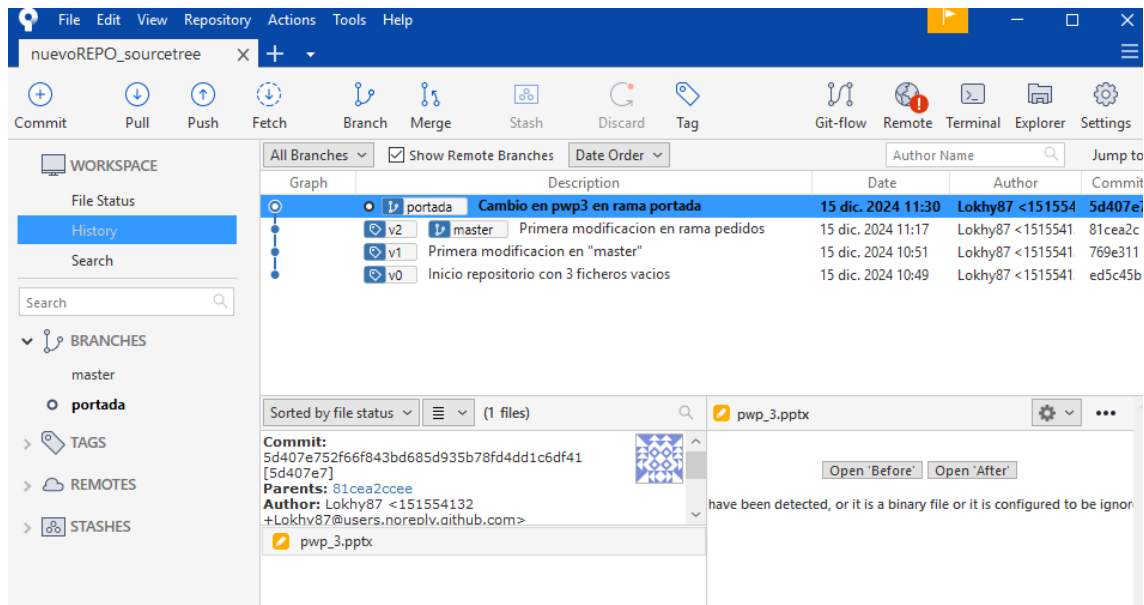


2. Desarrollo de la nueva página principal o portada del sitio (4 puntos)

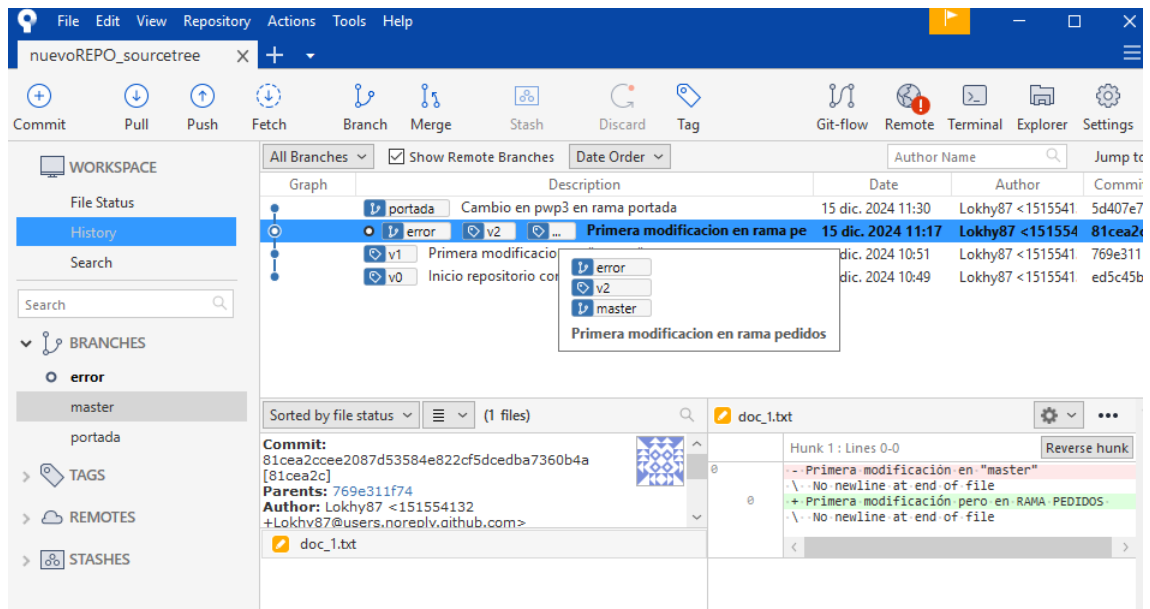
- Vamos a suponer que el equipo de desarrollo comienza ahora a realizar el desarrollo de la nueva portada web. Genera una nueva rama desde la última consolidación y llámala “portada”.



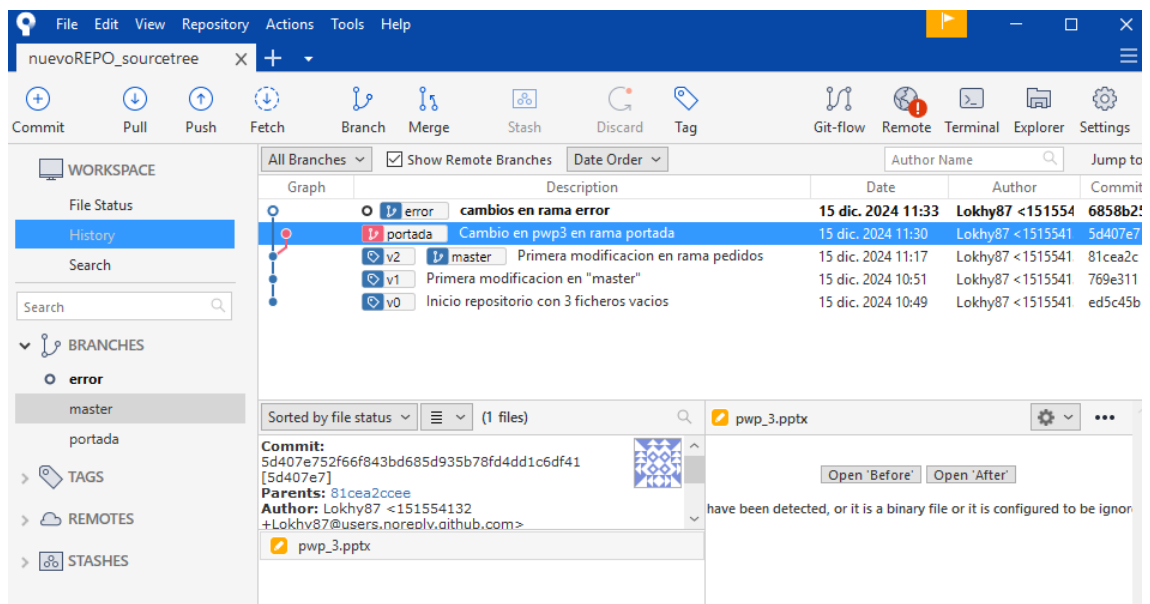
- Con la rama “portada” activa, haz cambios en algún fichero del directorio de trabajo y consolida los cambios.



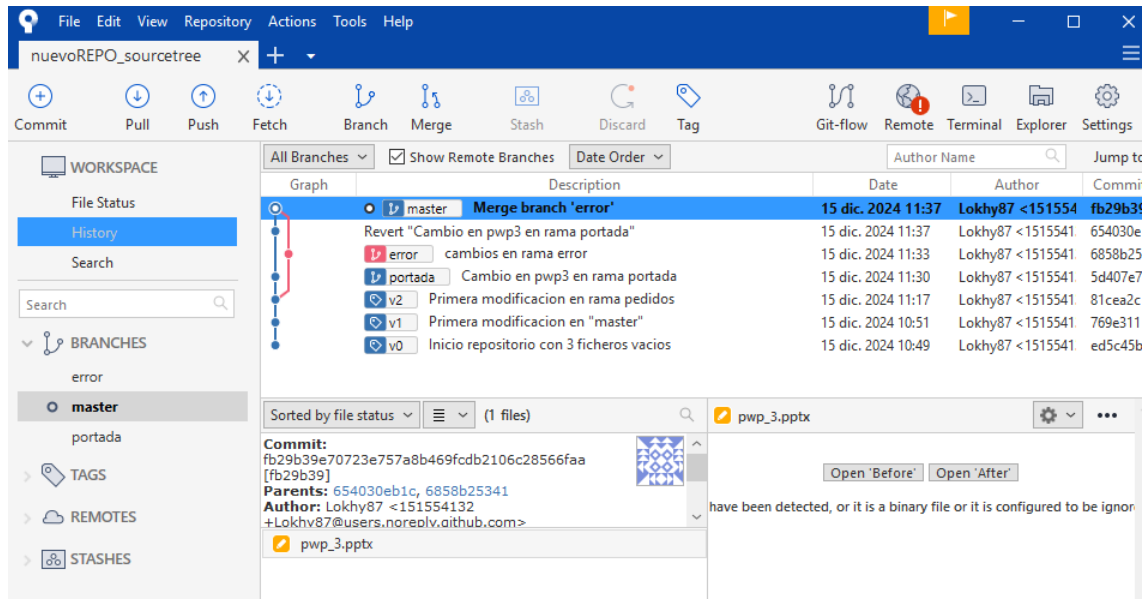
- En este momento, el equipo es informado de que debe dedicarse a resolver un ‘bug’ o error detectado en la rama “master”. Con la rama “master” activa, crea una nueva rama denominada “error”, desde la última consolidación de la rama “master”.



- Con la rama “error” activa, realiza un cambio en la misma línea o instrucción del mismo fichero que habías modificado justo antes en la rama “portada” y consólidalo en el repositorio.



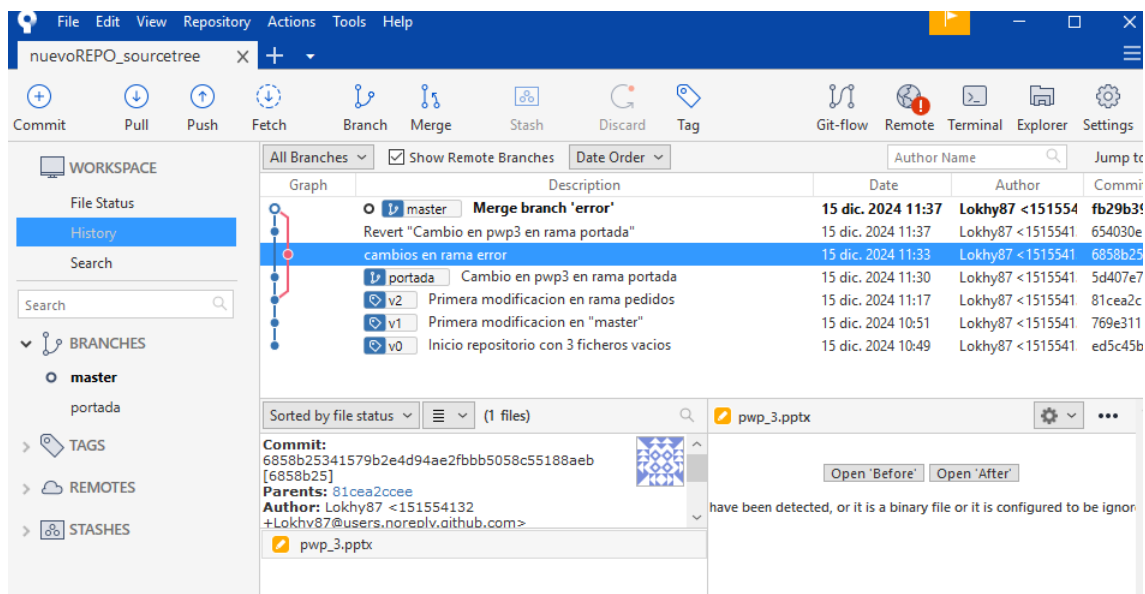
- El error se da por resuelto. Integra los cambios que has hecho para resolver el error, en la rama “master”.



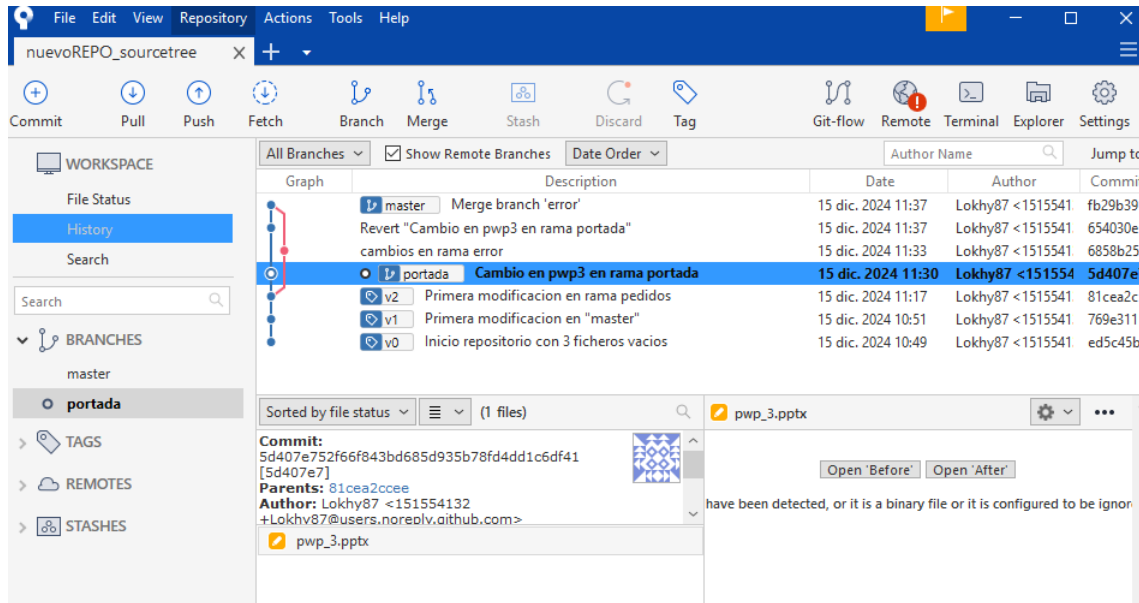
- Explica qué tipo de fusión has llevado a cabo.

Es un **merge no fast-forward**, es decir, una fusión en la que se conserva el historial de las 2 ramas, “master” y “error”

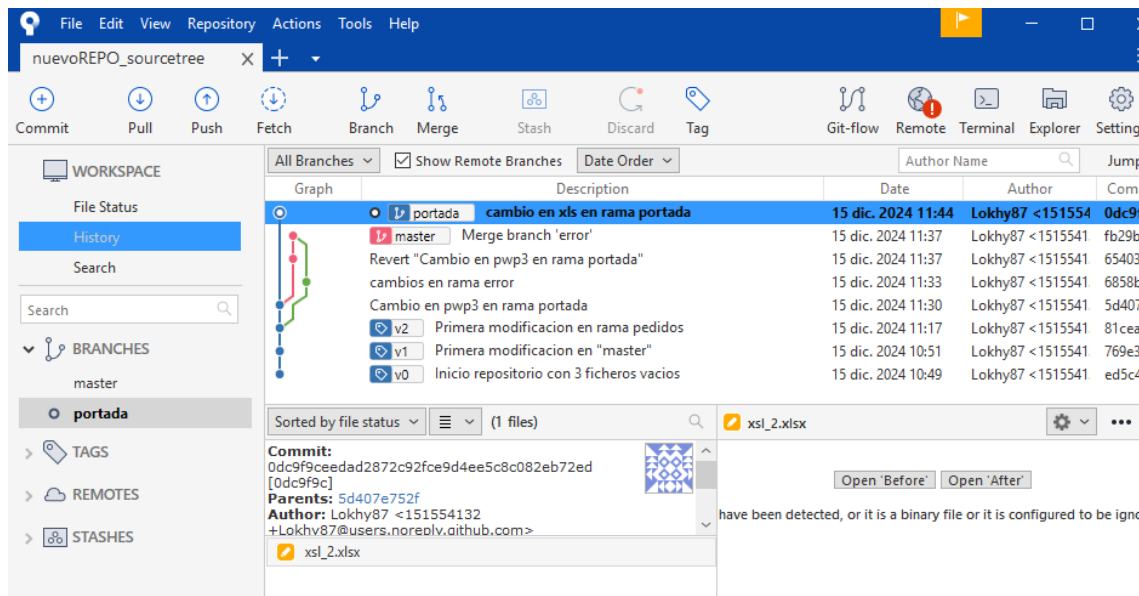
- Llegados a este punto, como hemos terminado de resolver el error y hemos integrado o fusionado los cambios en la rama “master”, podemos borrar la rama “error”. Bórrala y continúa los pasos.



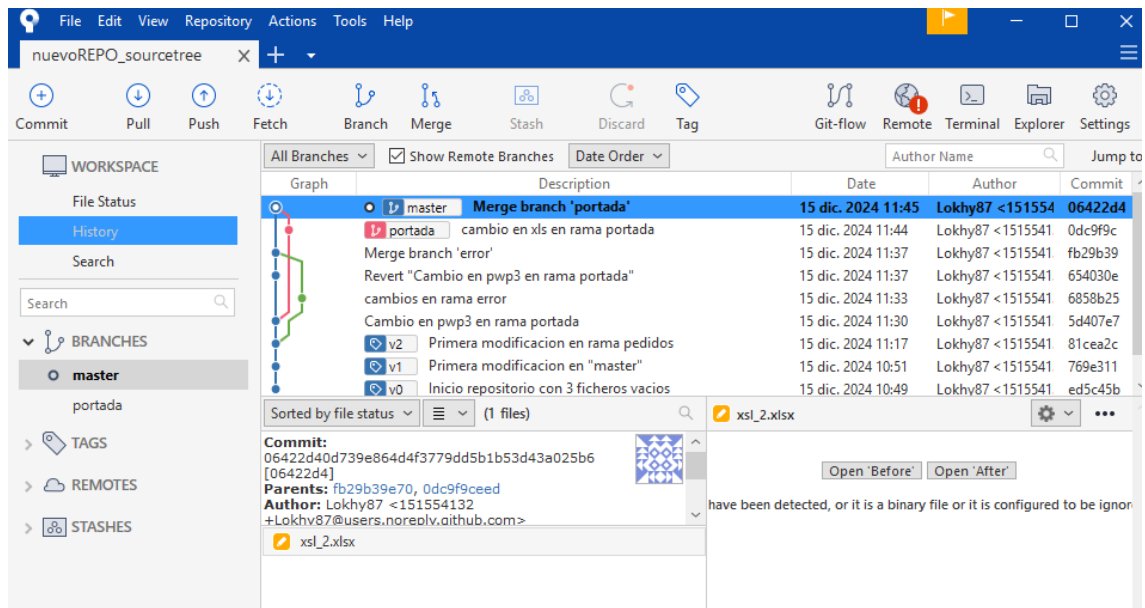
- Haz checkout de nuevo a la rama “portada”, es decir sitúa la rama “portada” como rama activa del repositorio (recuerda que hacer checkout entre ramas no es lo mismo que hacer checkout a una consolidación anterior dentro de una misma rama).



- Realiza alguna modificación en algún fichero del directorio de trabajo (en este caso, mejor si modificas algo que no se haya modificado en las ramas “error” ni “master”). Consolida los cambios.



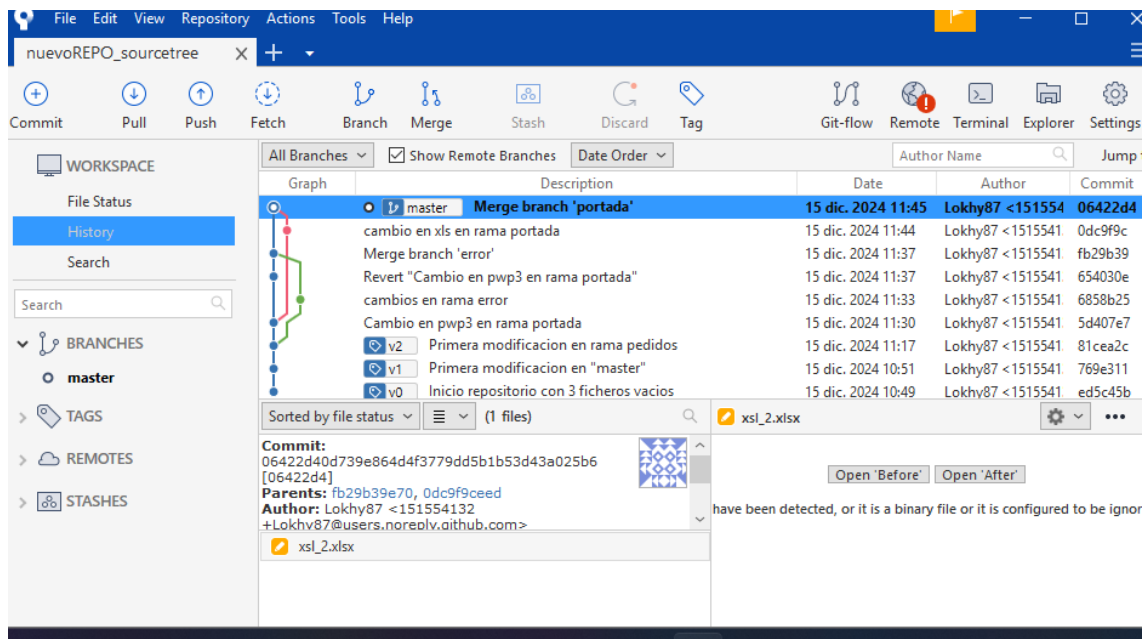
- Integra los cambios en la rama “master” mediante una fusión.



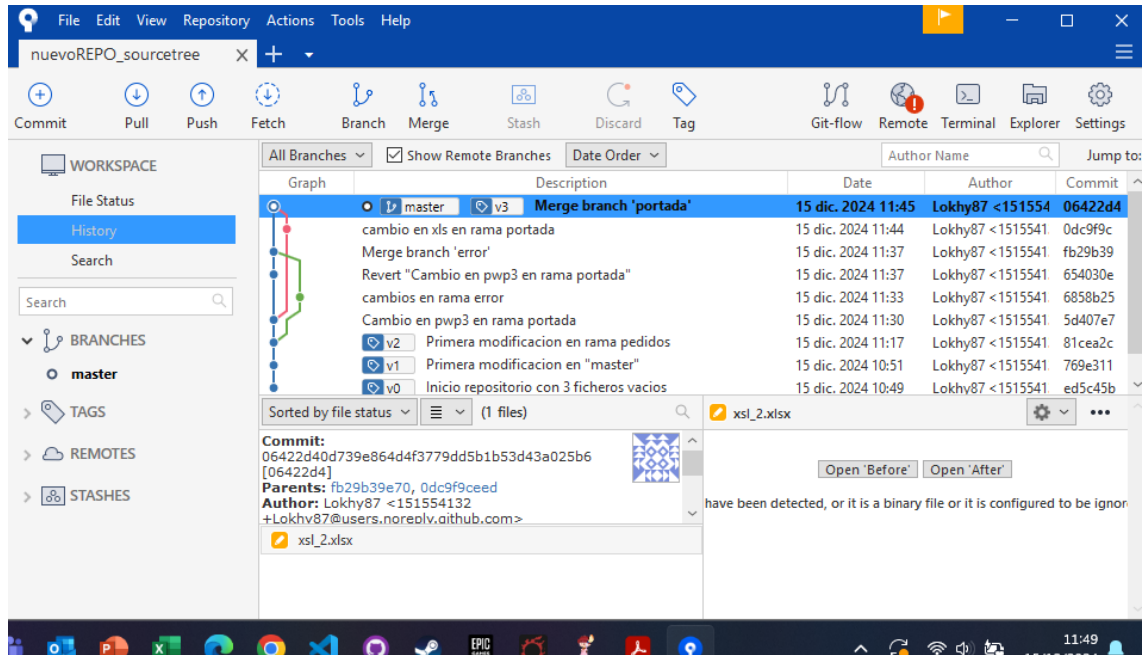
- Explica el tipo de fusión que se haría en este caso. Si ha sucedido algo particular, explícalo con detalle y explica también como lo gestionarías.

La fusión es **merge no fast-forward** debido a desarrollos paralelos en las ramas. Si hubiese habido conflicto hubiera resuelto manualmente

- Llegados a este punto, como hemos terminado la portada y la hemos integrado junto con la corrección del error en la rama “master”, podemos borrar la rama “portada”. Bórrala y continúa los pasos.

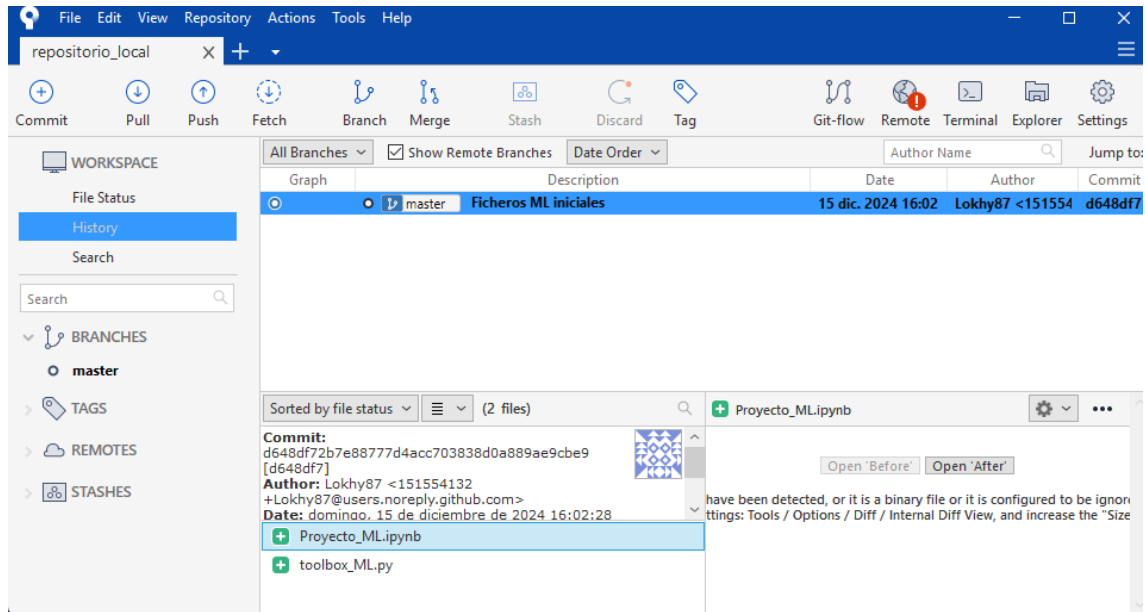


- Etiqueta la última consolidación como v3.

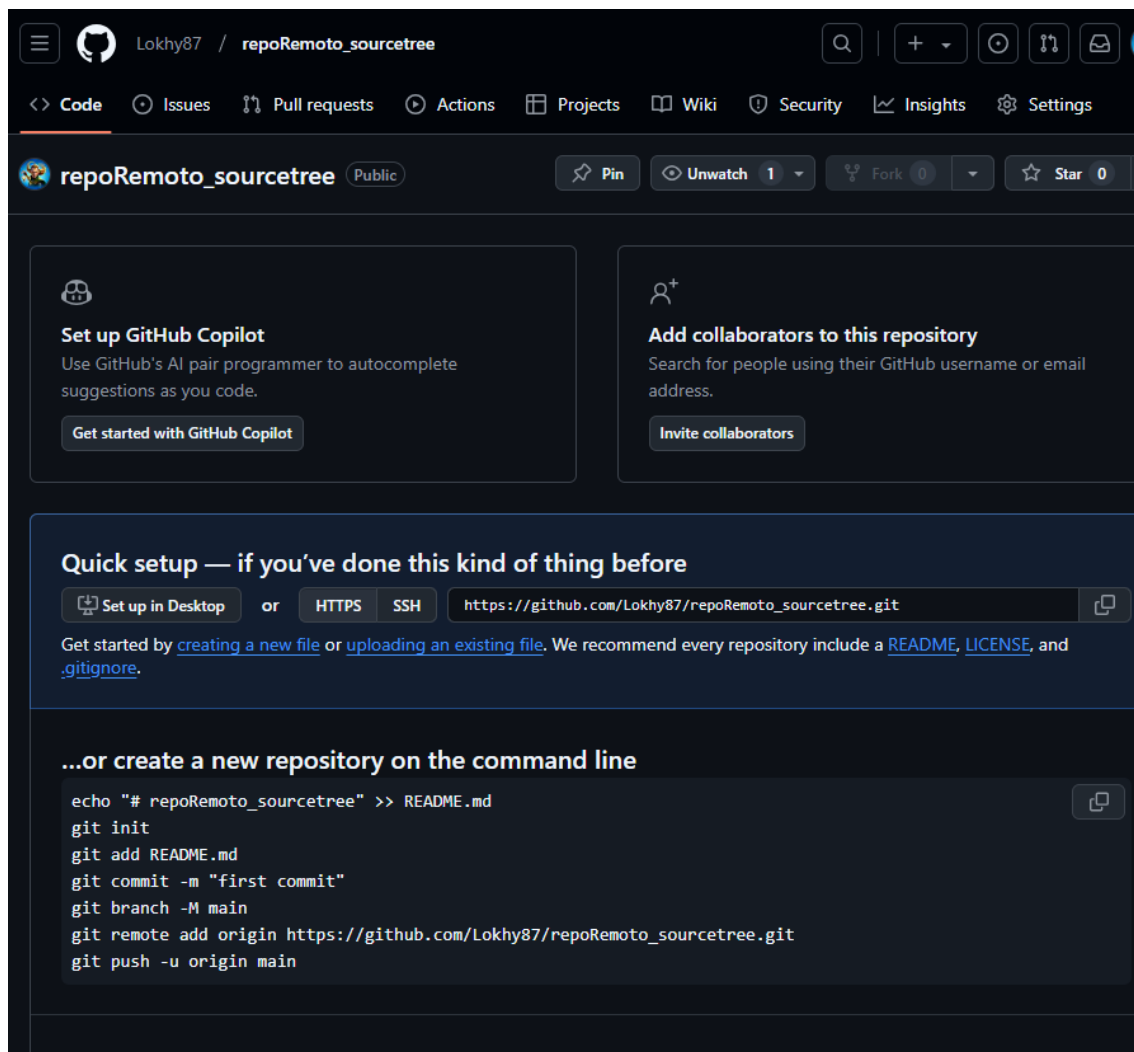


3. Repositorio remoto desde repositorio local (1.5 puntos)

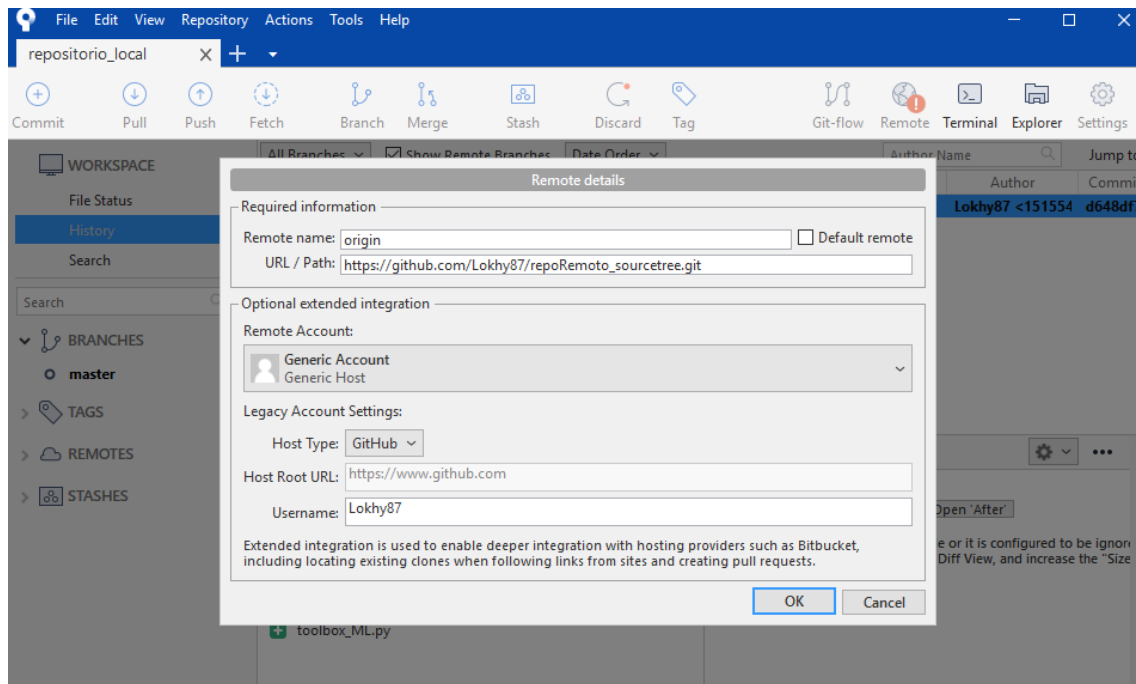
- Crea un repositorio local en tu máquina con varios ficheros.



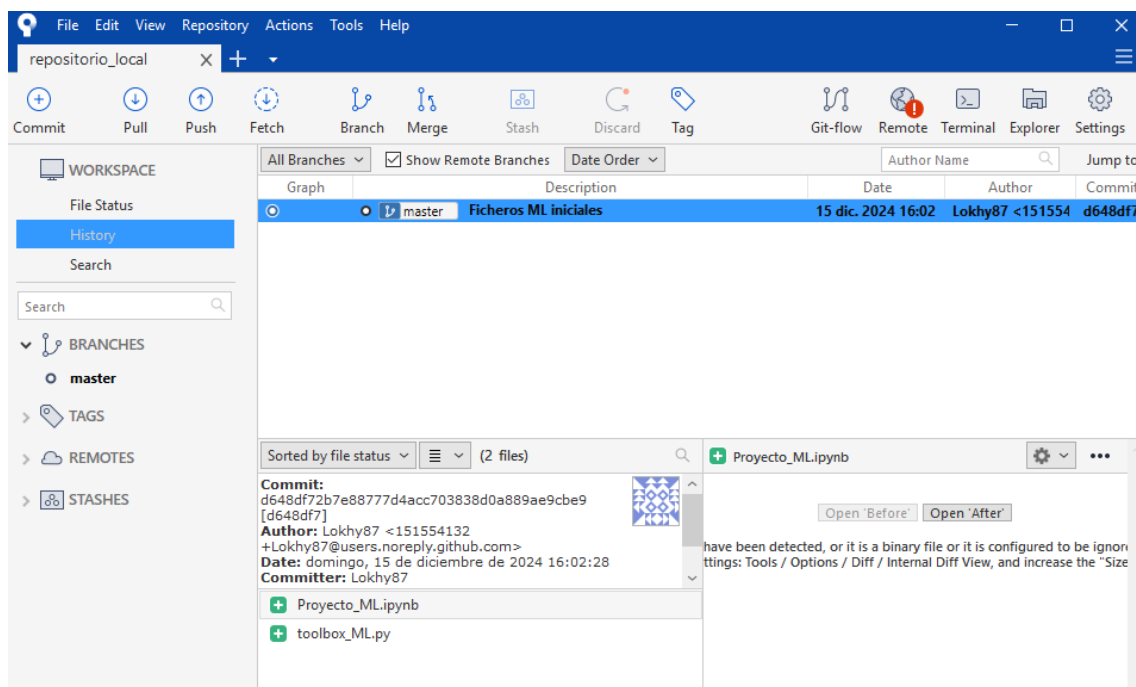
- Crea un repositorio remoto, por ejemplo, en Github.



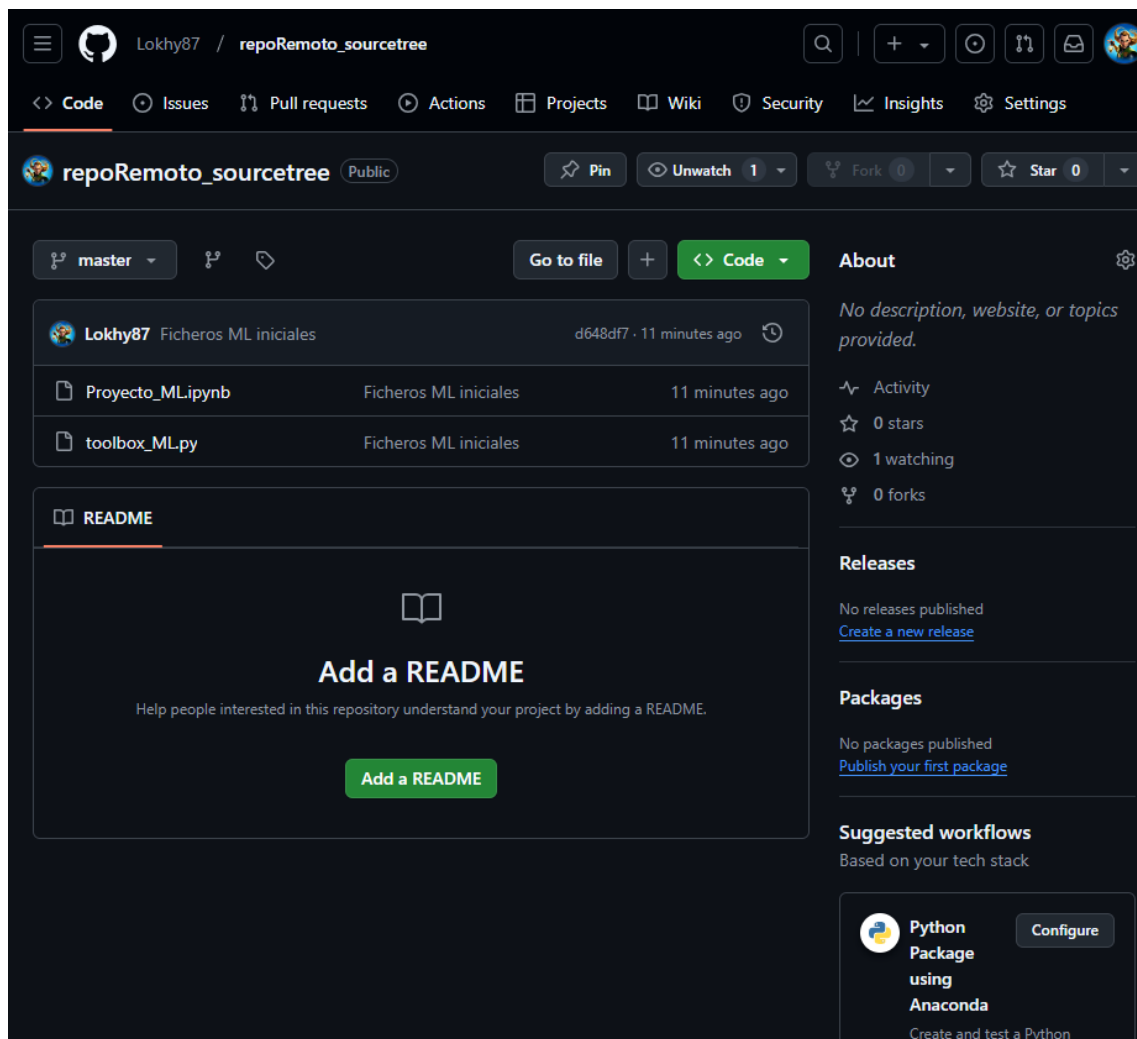
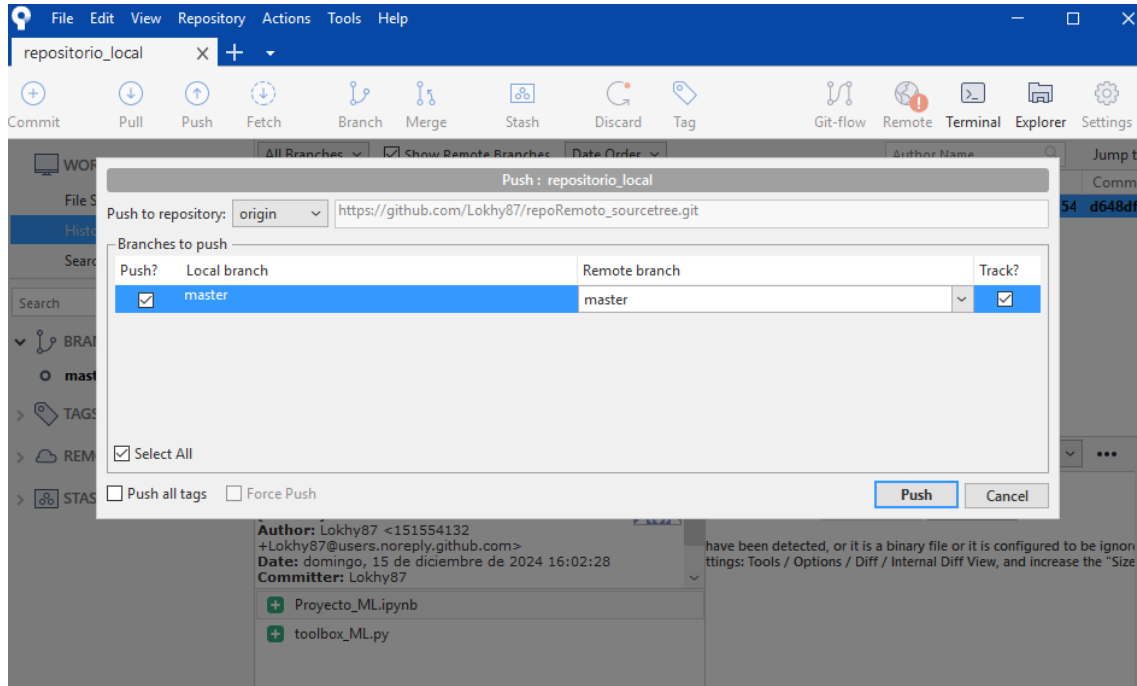
- Añade una conexión al repositorio remoto.



- Sincroniza el repositorio local con el repositorio remoto.

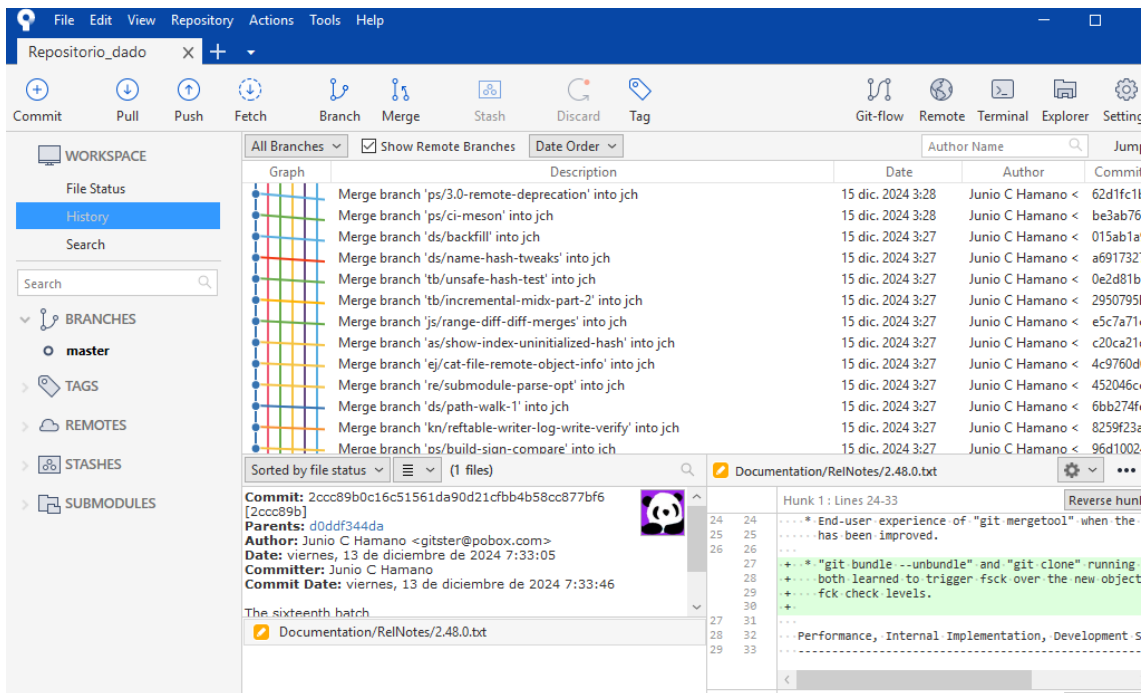
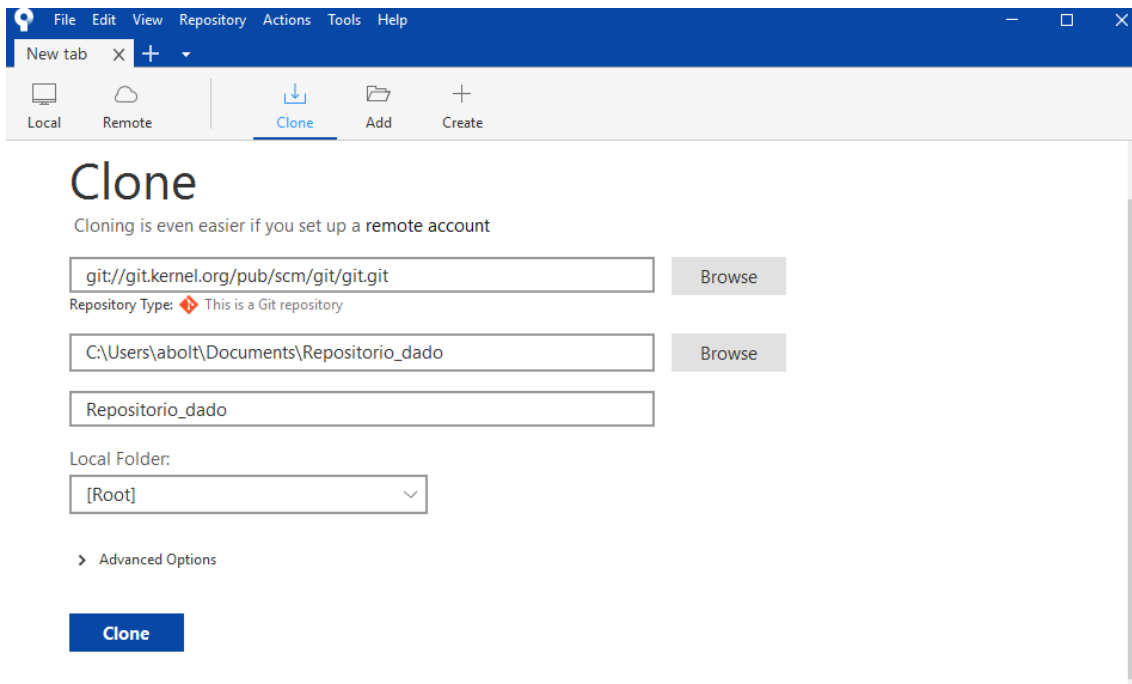


- Haz una consolidación en el repositorio local y después súbela al repositorio remoto.



4. Repositorio local desde repositorio remoto (1.5 puntos)

- Inicia un repositorio local desde el siguiente repositorio remoto existente: `git://git.kernel.org/pub/scm/git/git.git`



- Explica cómo integrarías tus modificaciones desde tu repositorio local en un repositorio remoto, en el que otras personas han ido sincronizando sus modificaciones también desde la última vez que sincronizaste. Detalla el ciclo de trabajo.

Actualización del repositorio local - Antes de modificar el repositorio remoto, debo asegurarme de que el local está actualizado con los cambios realizados.

Hago un “pull” con dichas modificaciones que hayan hecho.

Realizamos las modificaciones en local de nuestro trabajo.

Hacemos un commit con los cambios llevados a cabo en local y sincronizamos nuestros cambios con el repositorio remoto, haciendo un “push”.

Ciclo trabajo:

1. Sincronización local (pull)

Antes de realizar cualquier modificación en los archivos locales, sincronizar el repositorio local con el remoto.

Sourcetree: Pull – seleccionar rama remota y asegurar que coincida con rama local.

2. Modificación archivos locales.

Realizar cambios necesarios en el entorno local.

Sourcetree: Comprobar los archivos modificados, estarán en “Unstaged files”.

3. Commit de los cambios.

Seleccionar los archivos modificados y pasarlos al “Stage Selected” para meterlos al área de preparación (“staging area”).

Escribir la descripción en el texto de “commit”.

Clic en “Commit” para consolidar cambios en local.

4. Sincronización remota.

Hecho el “commit” subir los cambios al repositorio remoto.

Sourcetree: Asegurar los cambios y hacer el “push” para sincronizar los cambios con el repositorio remoto.

5. Gestión conflictos (si es necesario)

Si al realizar un “pull” o “push” se detectan conflictos (archivos modificados por otros colaboradores al a vez)

Sourcetree: Seleccionar archivos conflictivos en el editor. Realizar los cambios y consolidar con un nuevo “commit”.

6. Comprobacion final

Después del “push” verificar en el repositorio remoto que los cambios han sido integrados correctamente.