

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

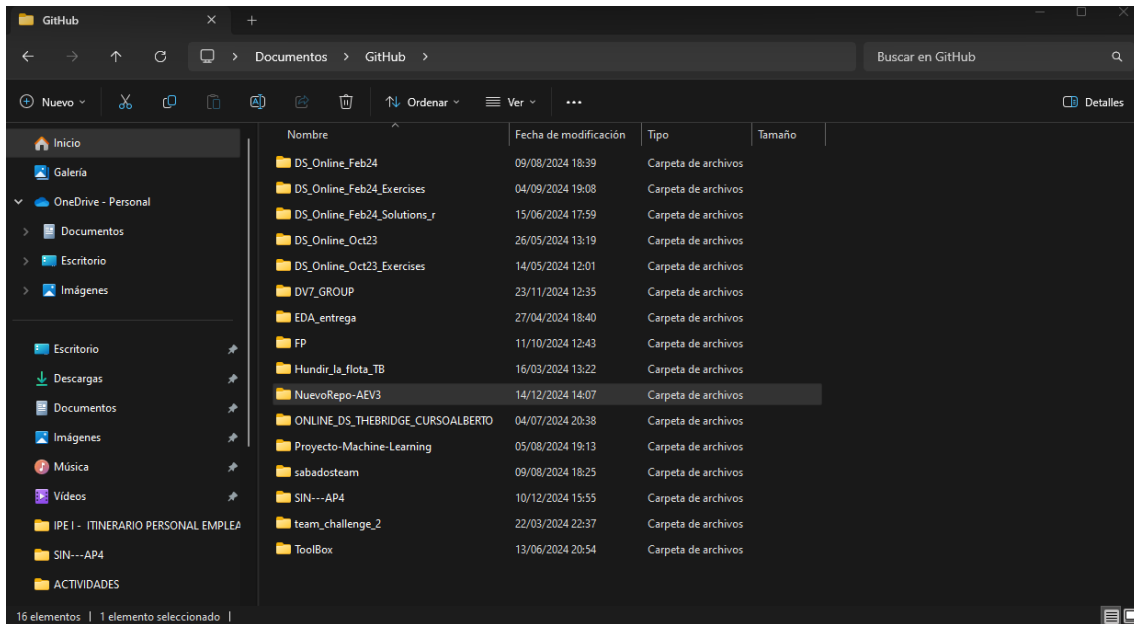
Puestos a disposición del alumno en el curso correspondiente del campus virtual Florida Oberta.

Detalle de la actividad

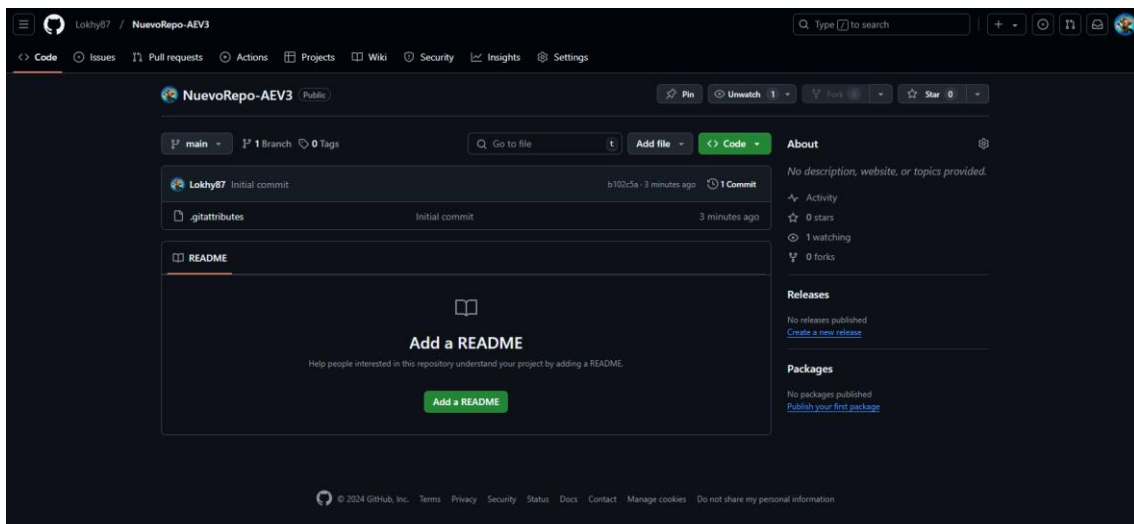
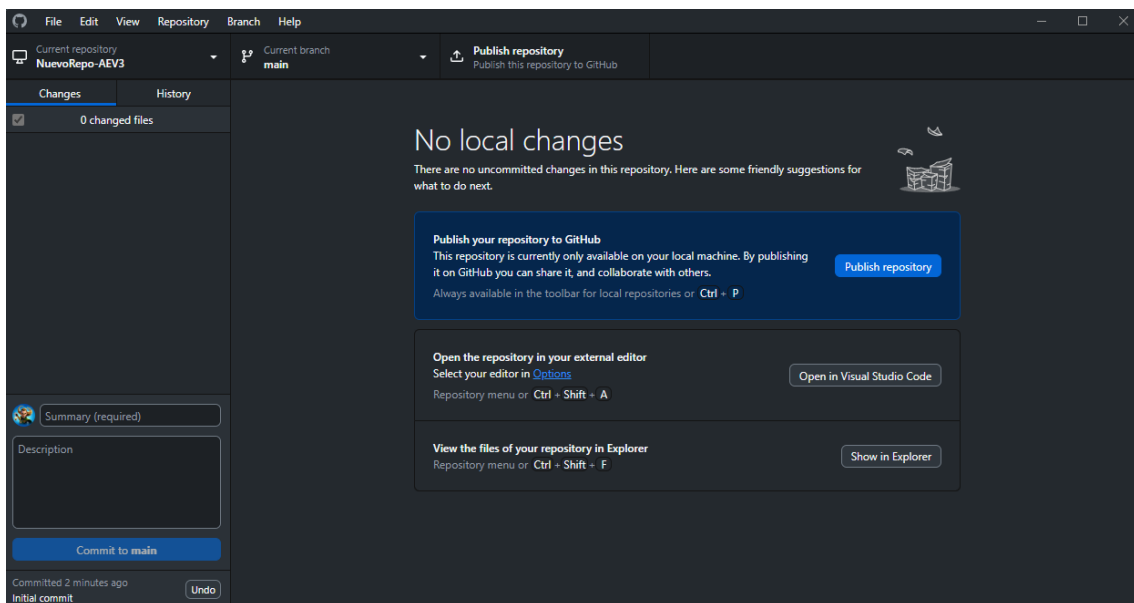
Supón que formas parte de un equipo de desarrollo de software y, entre tus funciones, está la de administrar el repositorio que se usa para gestionar la documentación de un proyecto, por ejemplo, la modernización de un portal web para la gestión de pedidos de clientes de una compañía de venta online:

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

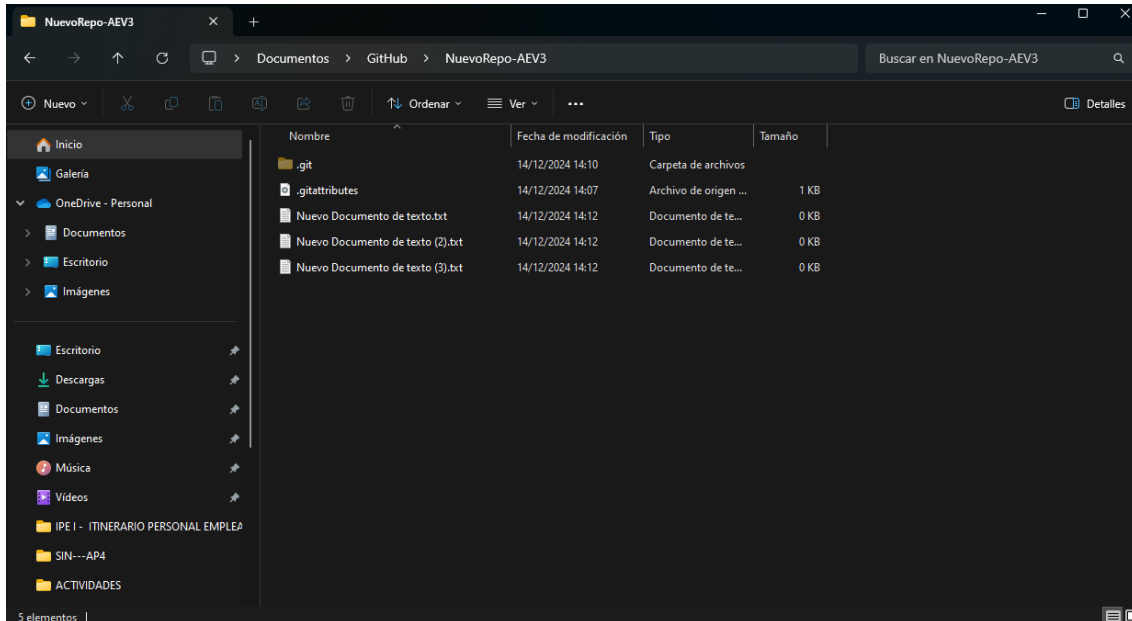
- Crea un nuevo directorio en tu sistema de ficheros local.



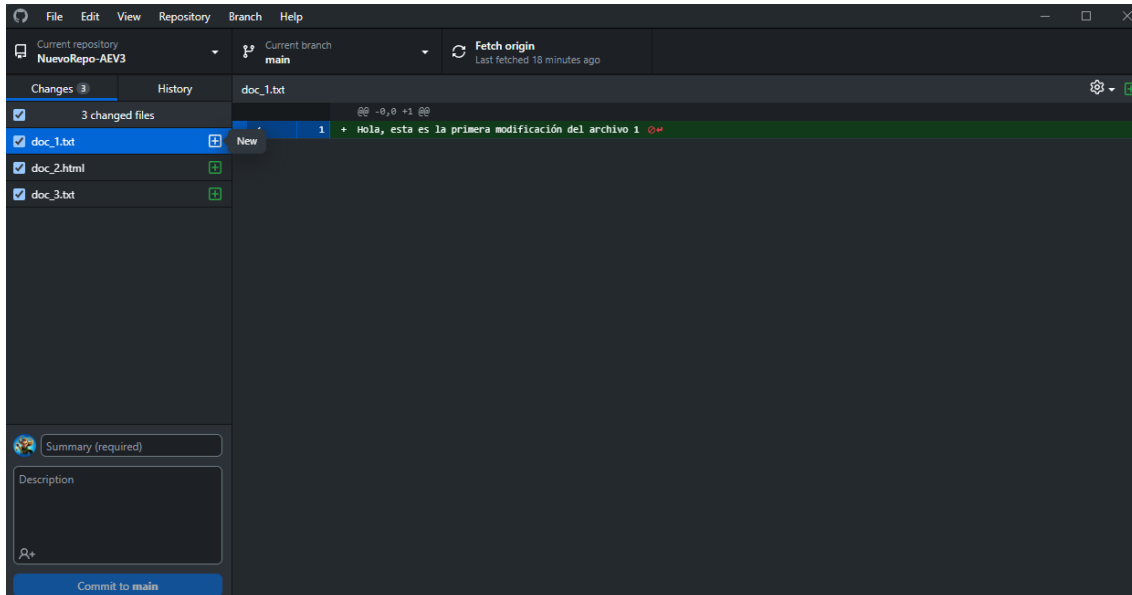
- Inicializa un repositorio en él.

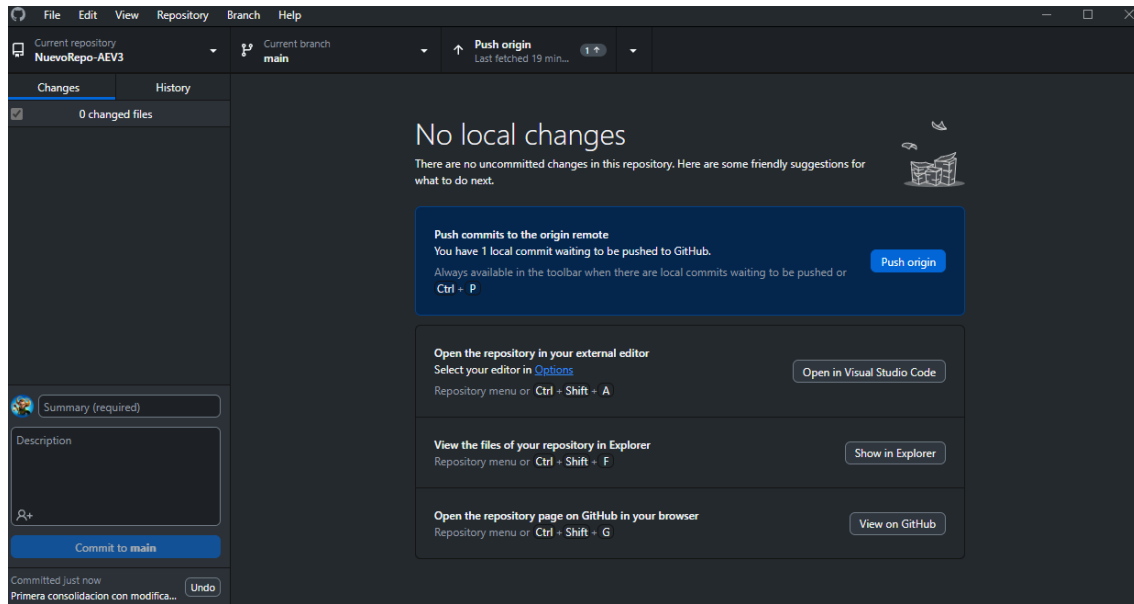


- Añade un mínimo de 3 ficheros vacíos al repositorio.

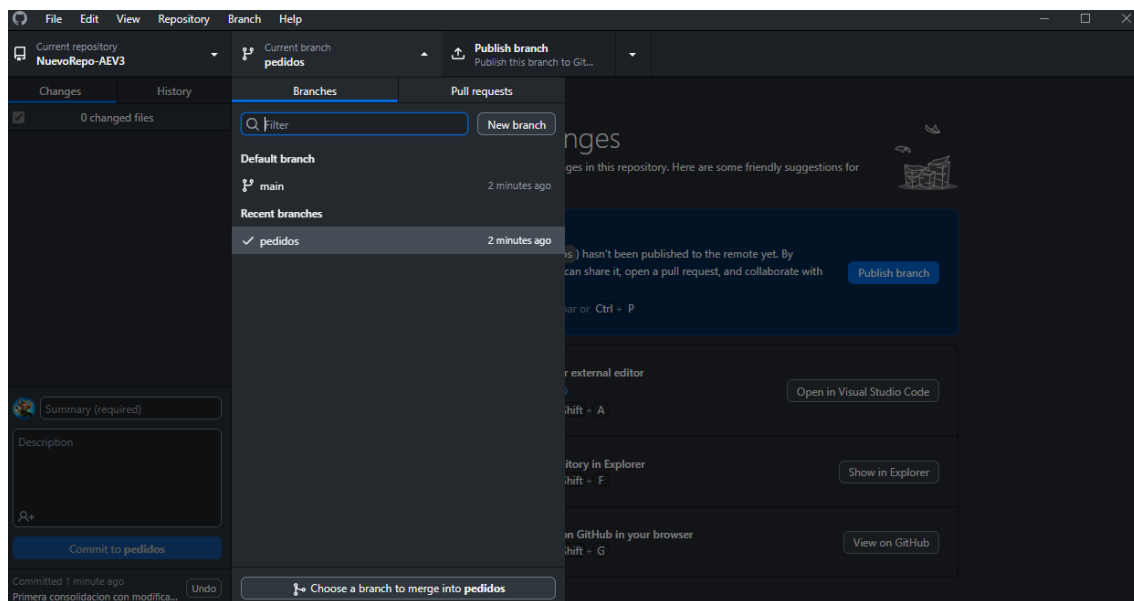


- Haz modificaciones en los ficheros y actualiza el repositorio mediante una consolidación.

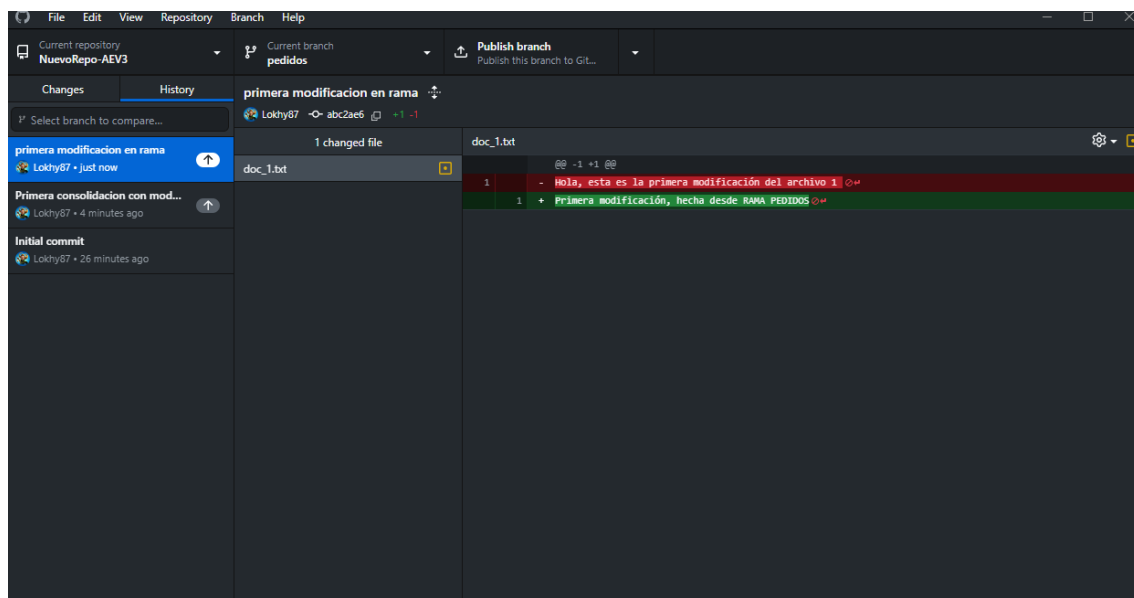
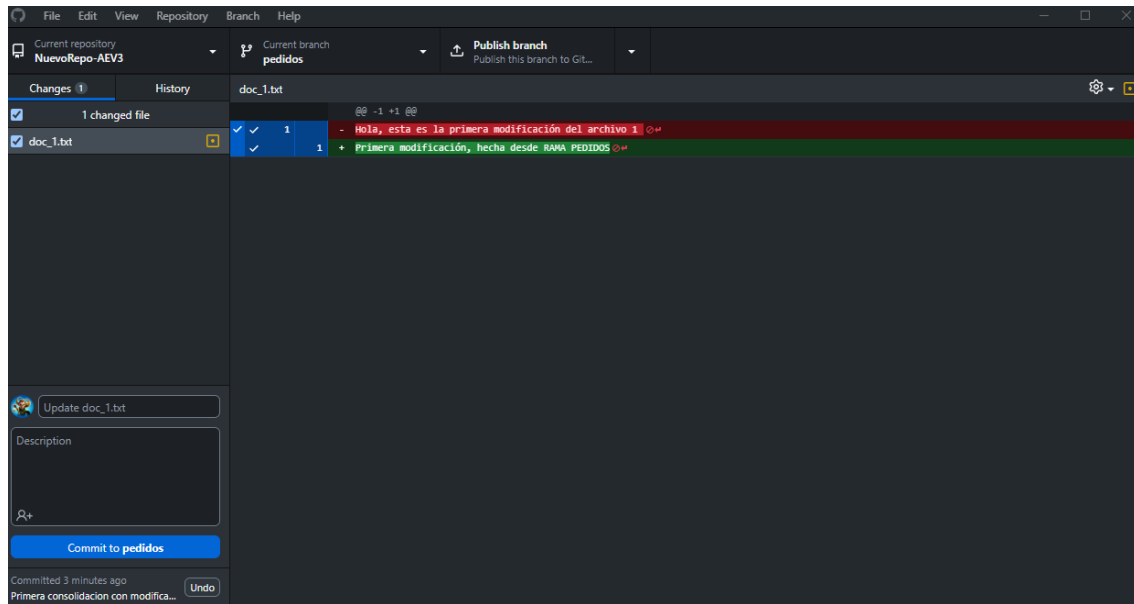




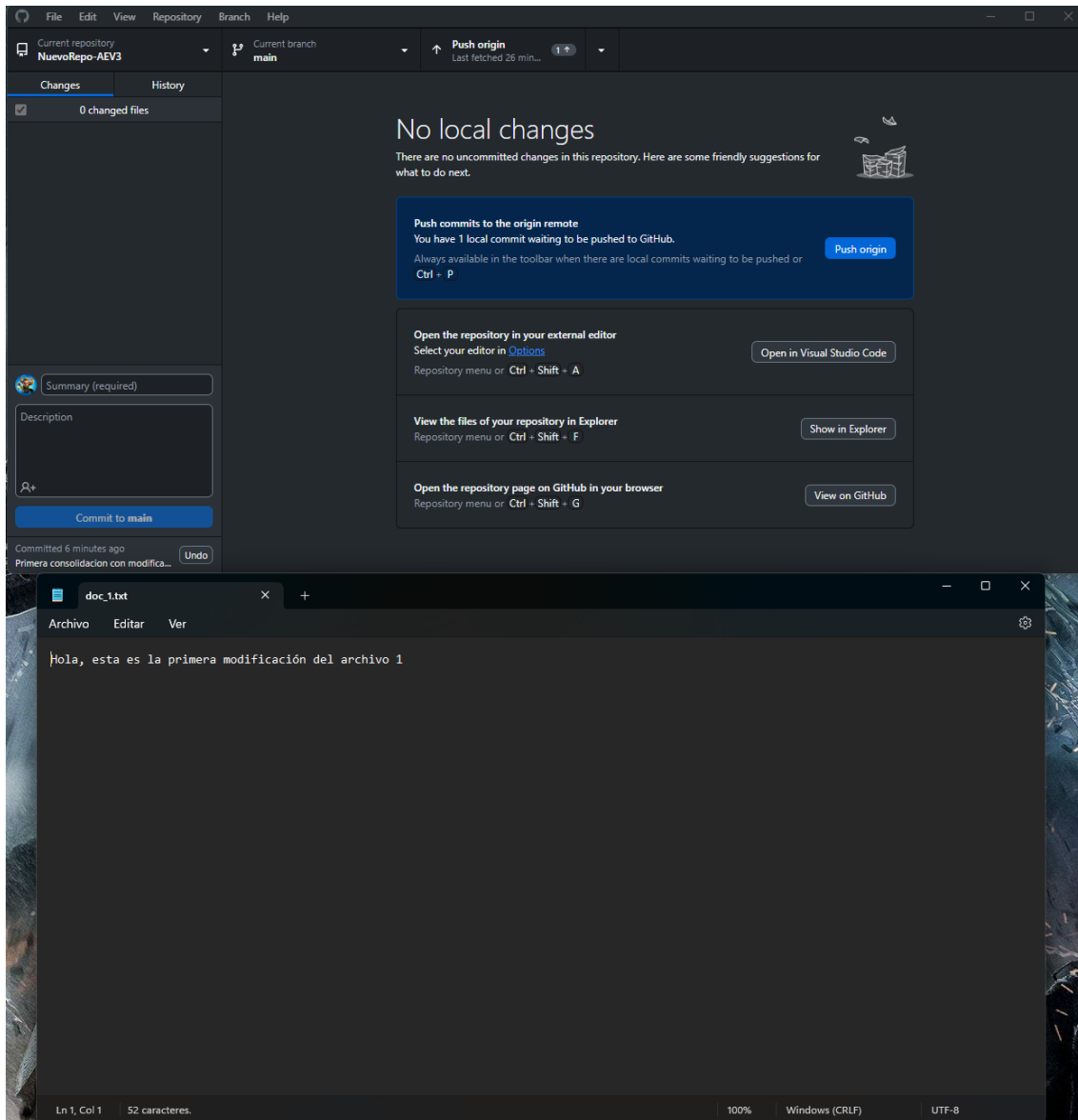
- El equipo va a comenzar a desarrollar la nueva pantalla de pedidos y va a requerir una línea de trabajo específica. Crea una rama para esta funcionalidad, por ejemplo, llámala “pedidos”.



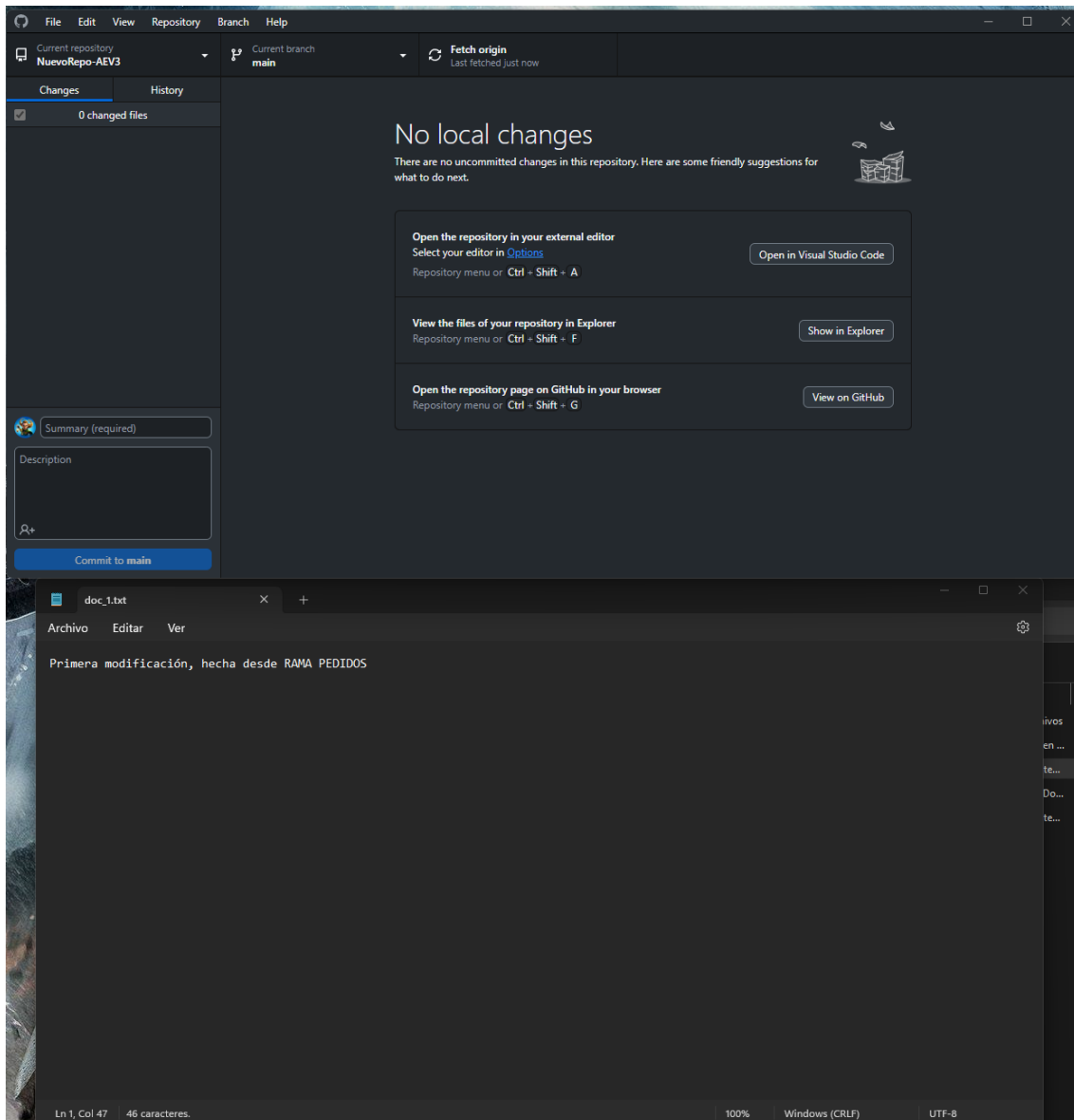
- Con la rama “pedidos” activa, realiza un cambio en algún fichero y consolida los cambios en dicha rama.



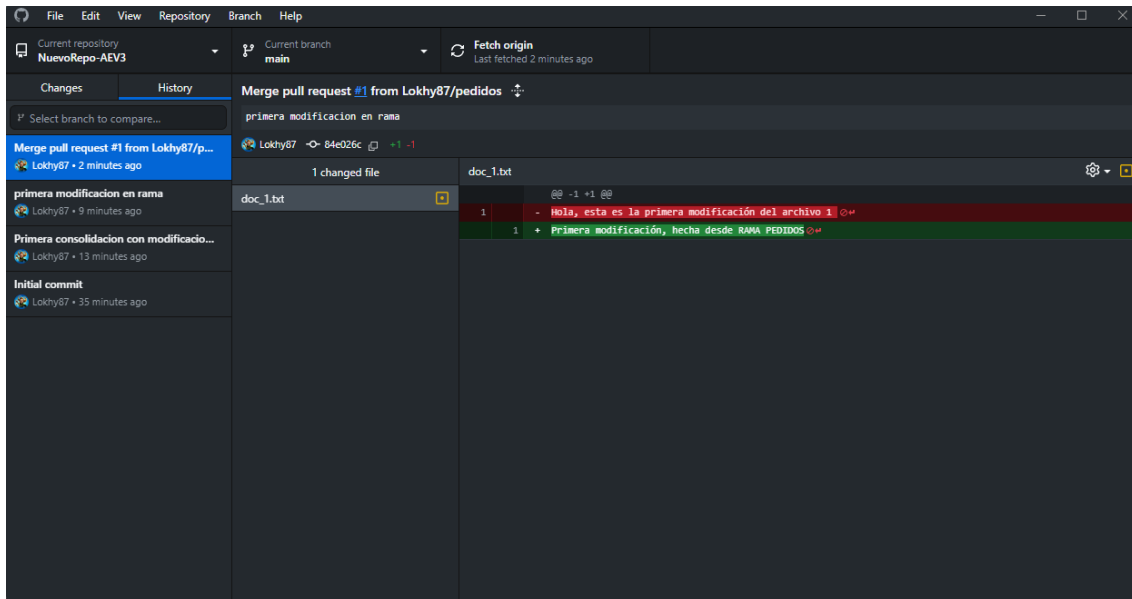
- Suponemos que hemos terminado de desarrollar la nueva pantalla de pedidos. Comprueba que los cambios realizados en la rama “pedidos”, no existen en los ficheros de la rama “master”.



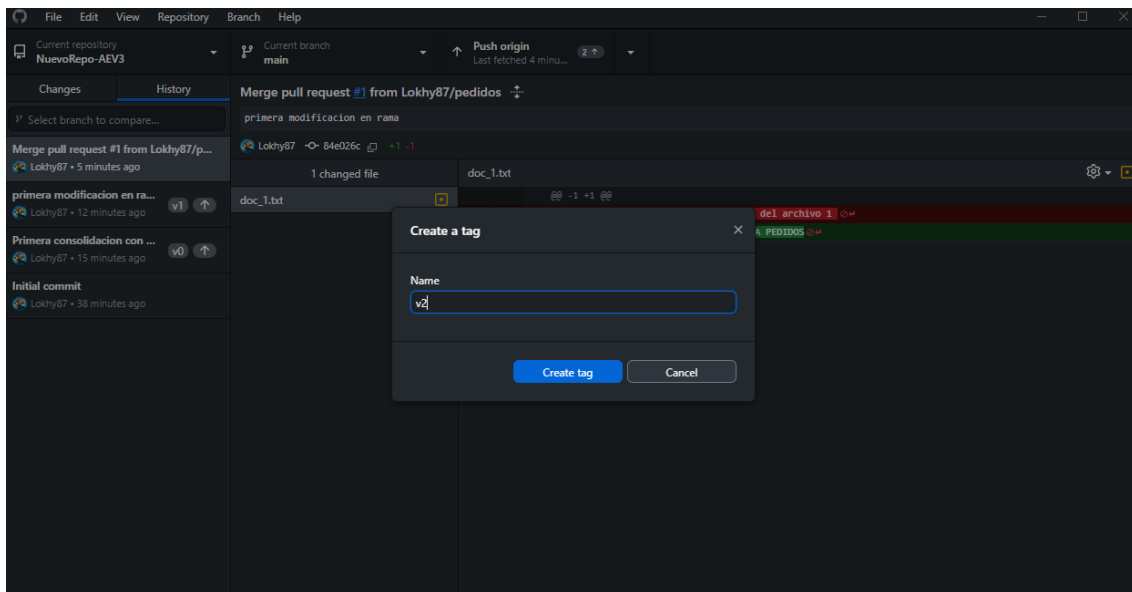
- Integra los cambios de la rama “pedidos” en la rama “master”.



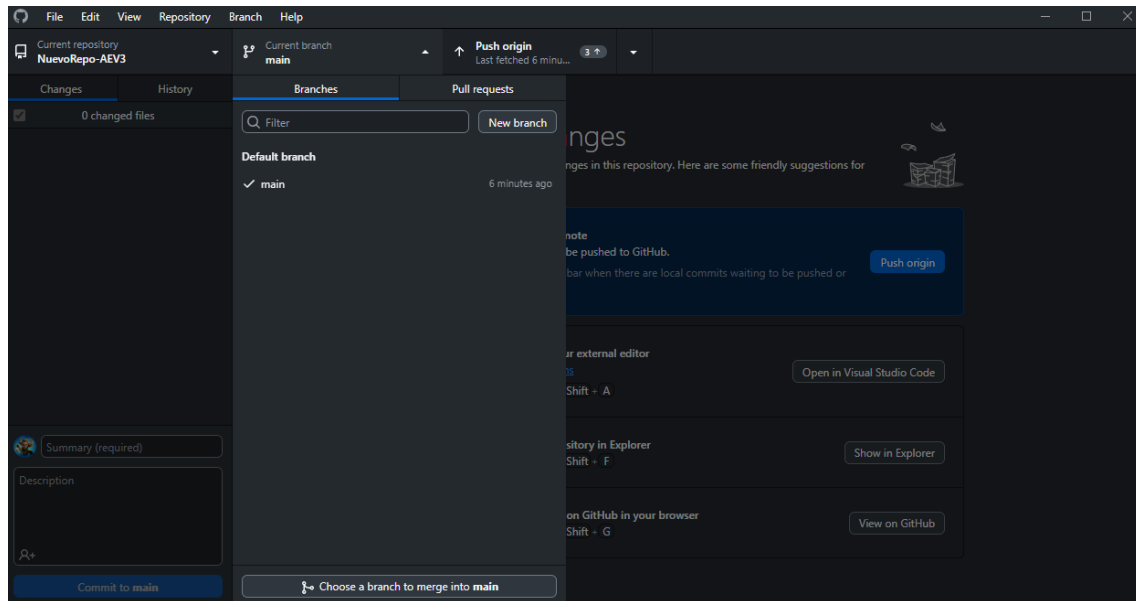
- Explica qué tipo de fusión has llevado a cabo.
- Después de integrar, comprueba si ahora existen en la rama “master”, los cambios que se habían realizado en la rama “pedidos”.



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

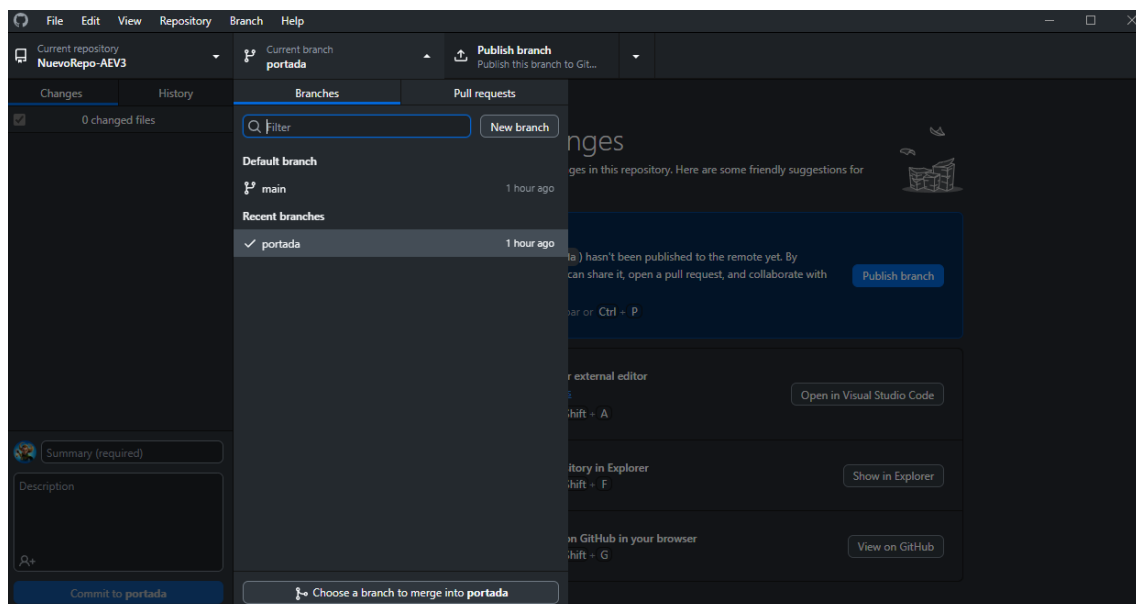


- Llegados a este punto, como hemos terminado la nueva pantalla de pedidos y la hemos integrado en la rama “master”, podemos borrar la rama “pedidos”, elimínala.

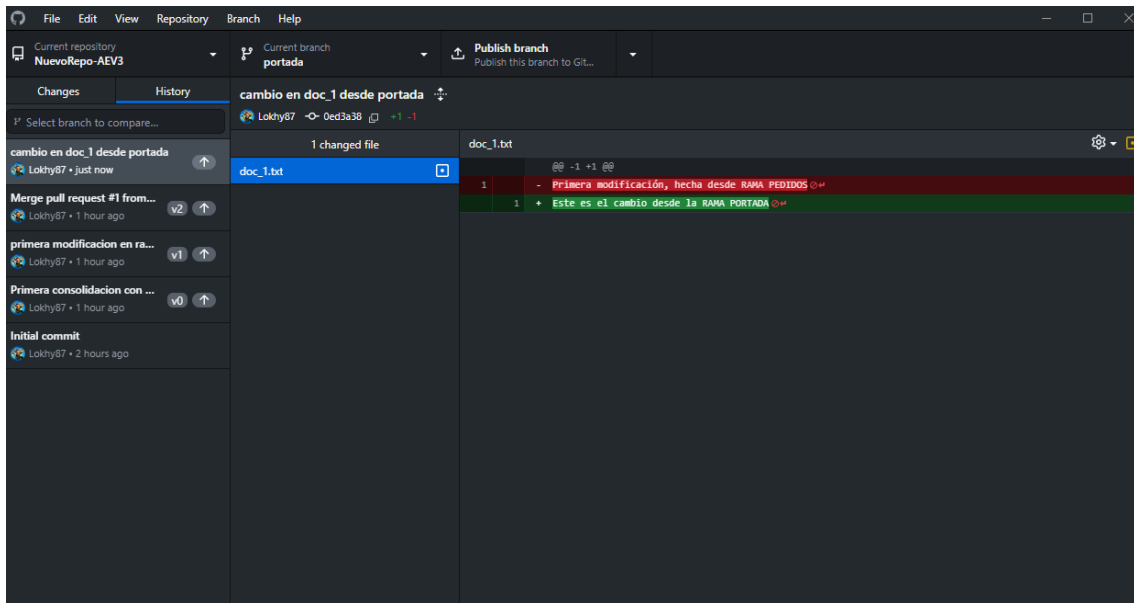
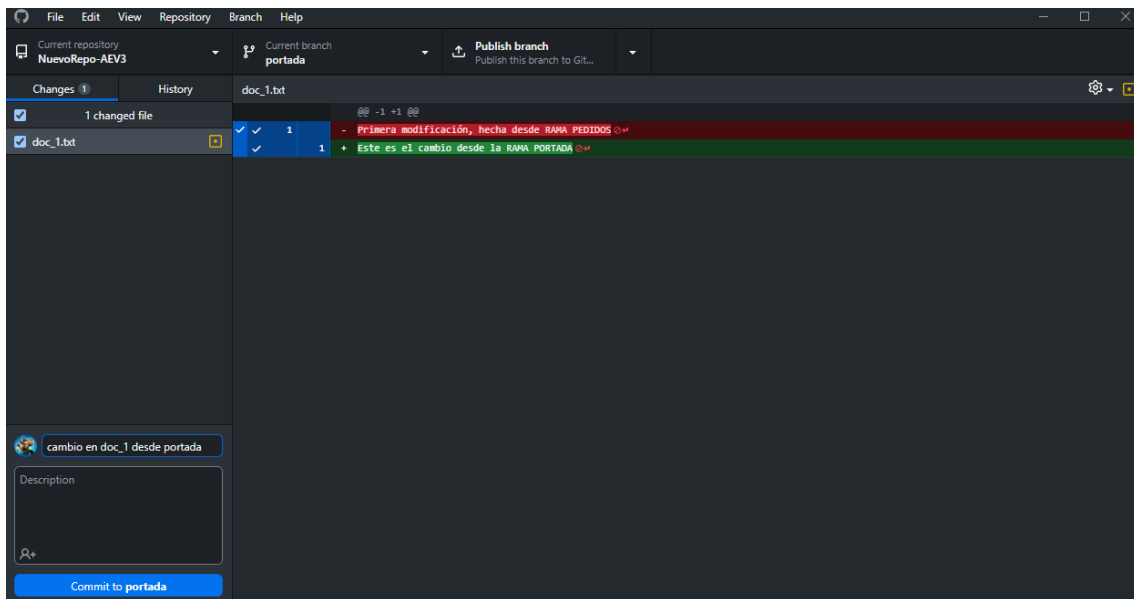


2. Desarrollo de la nueva página principal o portada del sitio (4 puntos)

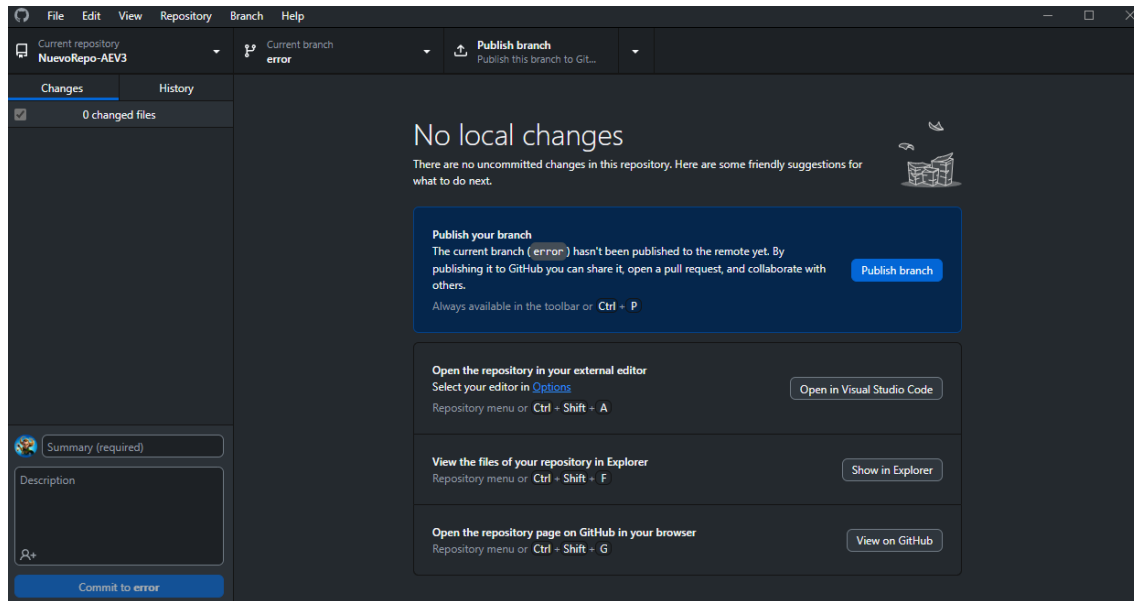
- Vamos a suponer que el equipo de desarrollo comienza ahora a realizar el desarrollo de la nueva portada web. Genera una nueva rama desde la última consolidación y llámala “portada”.



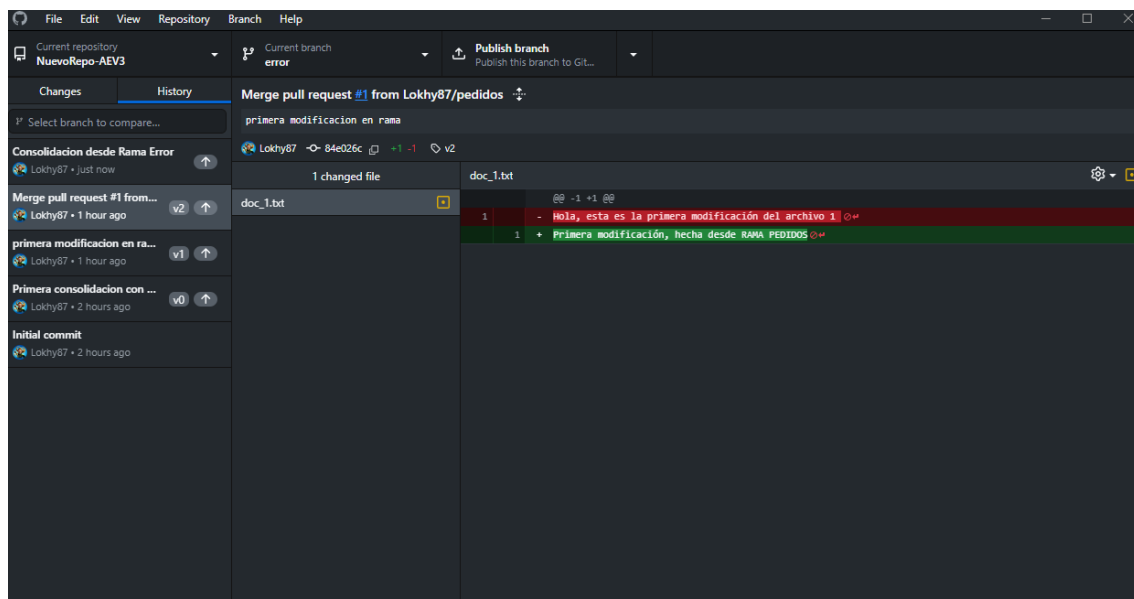
- Con la rama “portada” activa, haz cambios en algún fichero del directorio de trabajo y consolida los cambios.



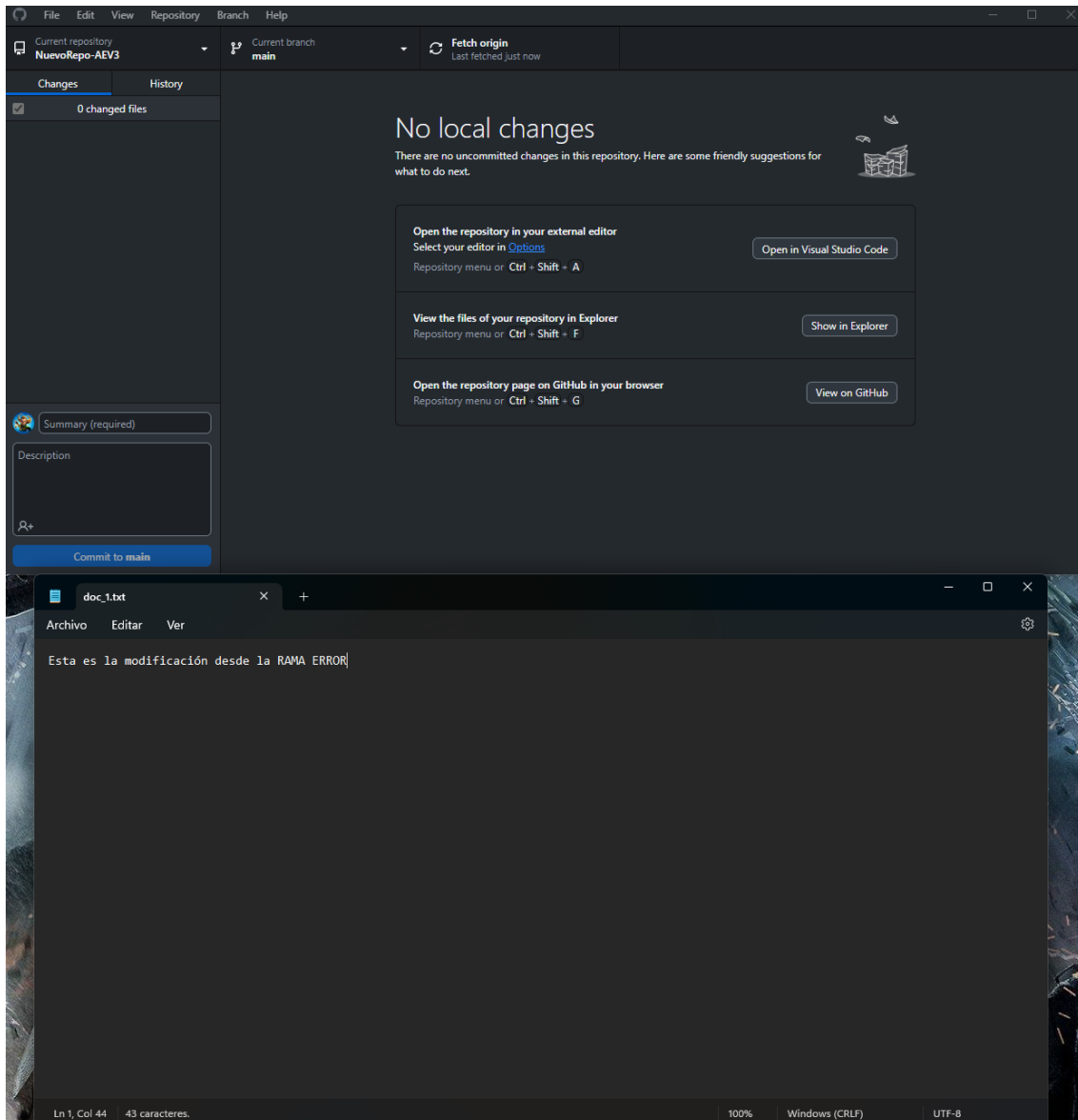
- En este momento, el equipo es informado de que debe dedicarse a resolver un ‘bug’ o error detectado en la rama “master”. Con la rama “master” activa, crea una nueva rama denominada “error”, desde la última consolidación de la rama “master”.



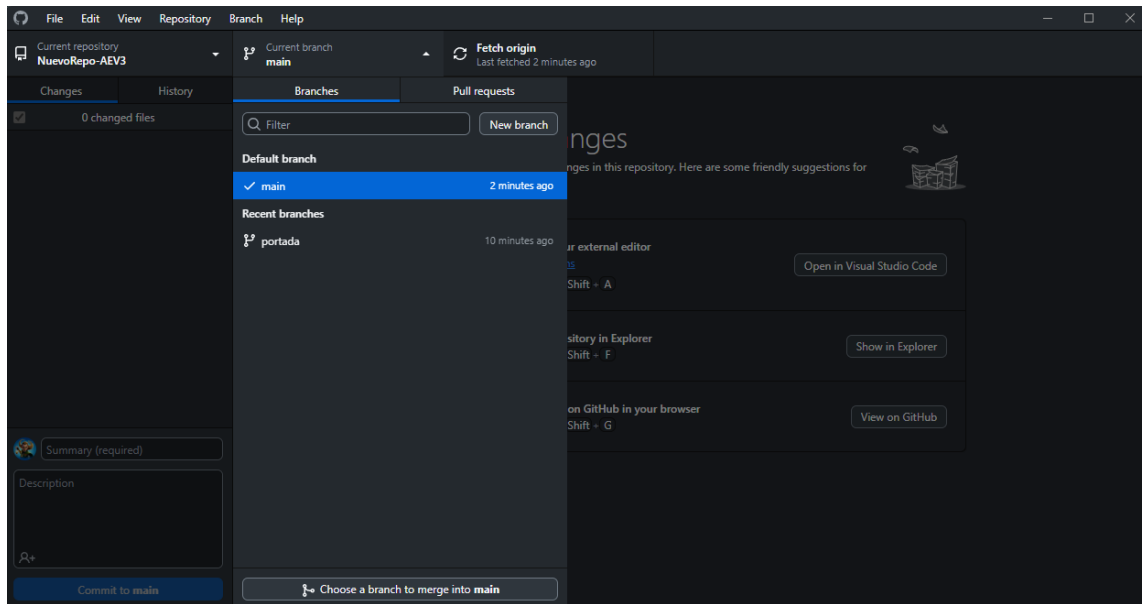
- Con la rama “error” activa, realiza un cambio en la misma línea o instrucción del mismo fichero que habías modificado justo antes en la rama “portada” y consolídalo en el repositorio.



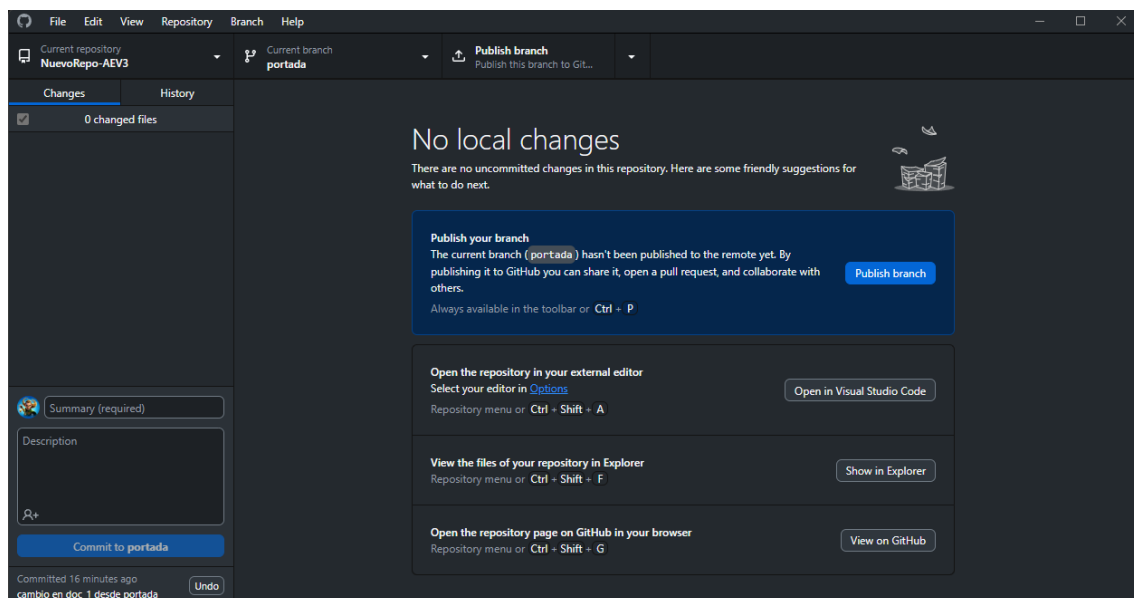
- El error se da por resuelto. Integra los cambios que has hecho para resolver el error, en la rama “master”.



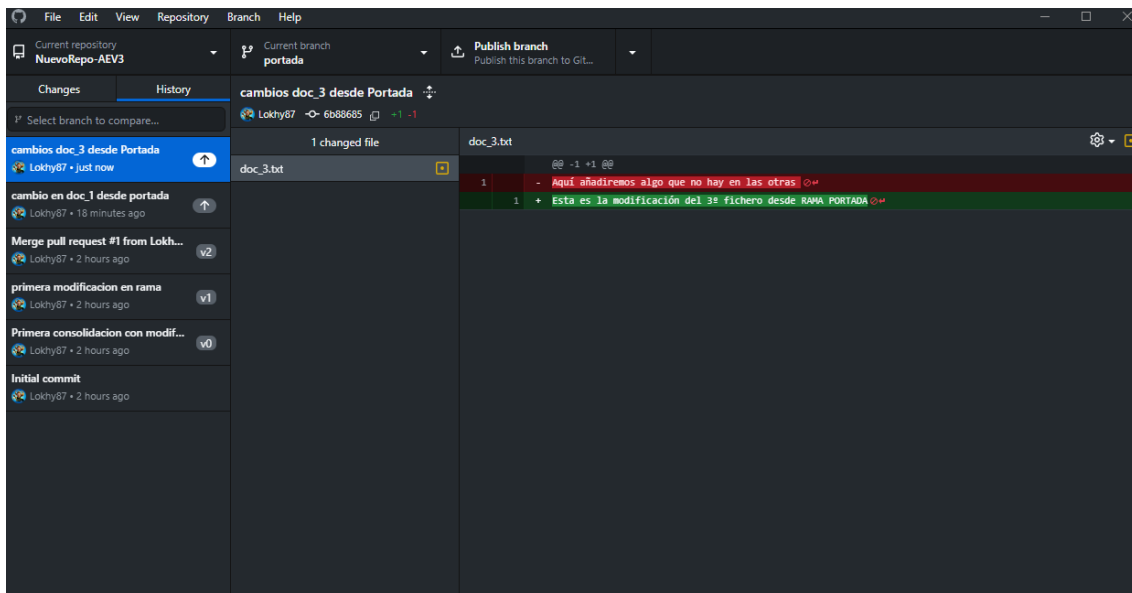
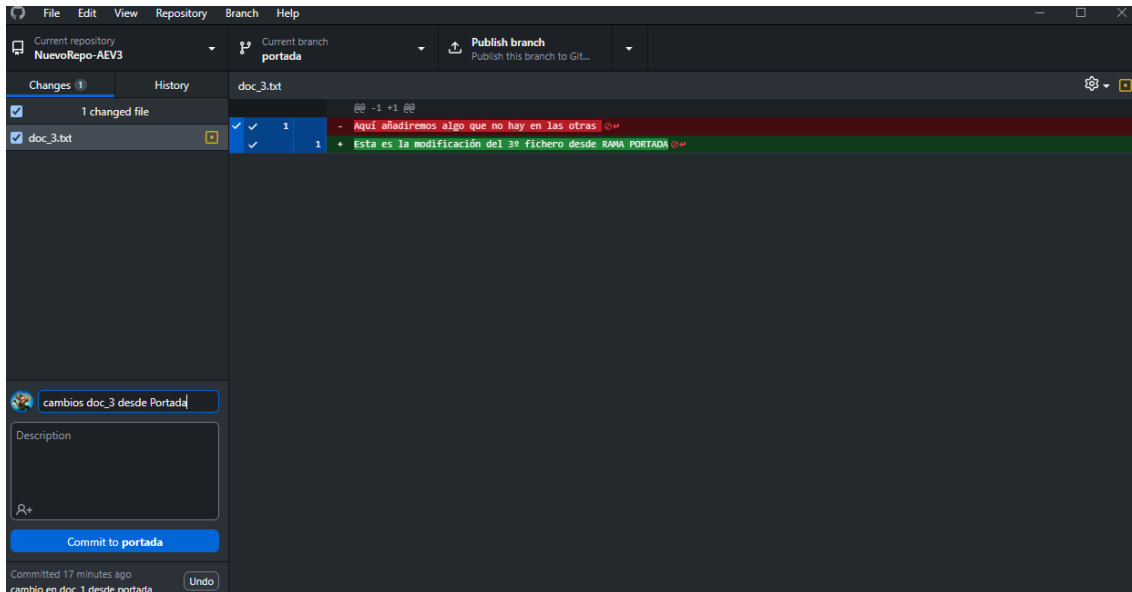
- Explica qué tipo de fusión has llevado a cabo.
- Llegados a este punto, como hemos terminado de resolver el error y hemos integrado o fusionado los cambios en la rama “master”, podemos borrar la rama “error”. Bórrala y continúa los pasos.



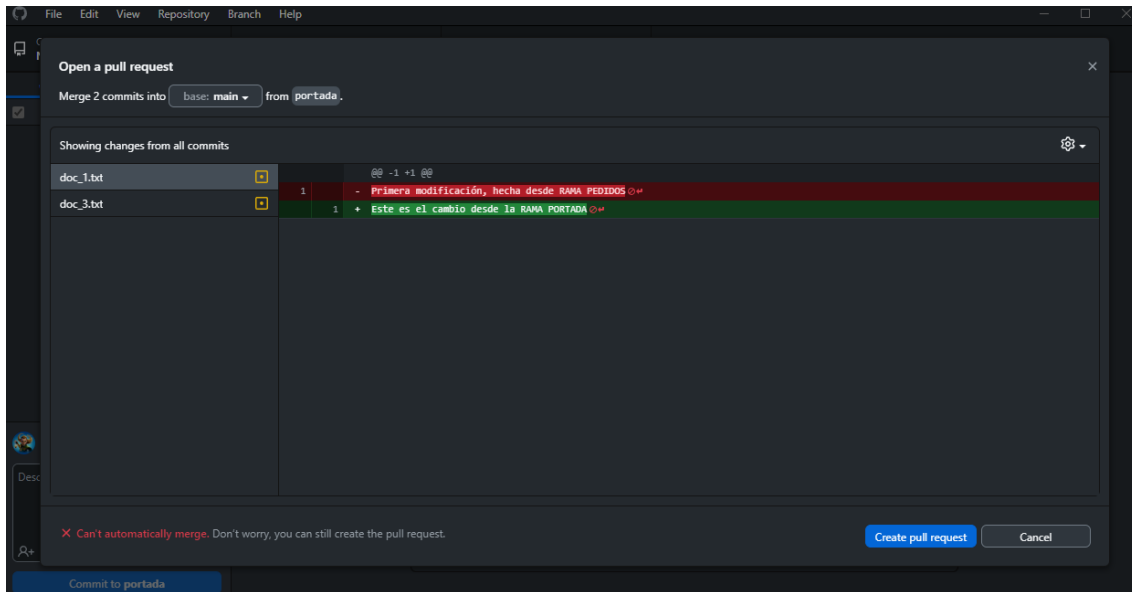
- Haz checkout de nuevo a la rama “portada”, es decir sitúa la rama “portada” como rama activa del repositorio (recuerda que hacer checkout entre ramas no es lo mismo que hacer checkout a una consolidación anterior dentro de una misma rama).



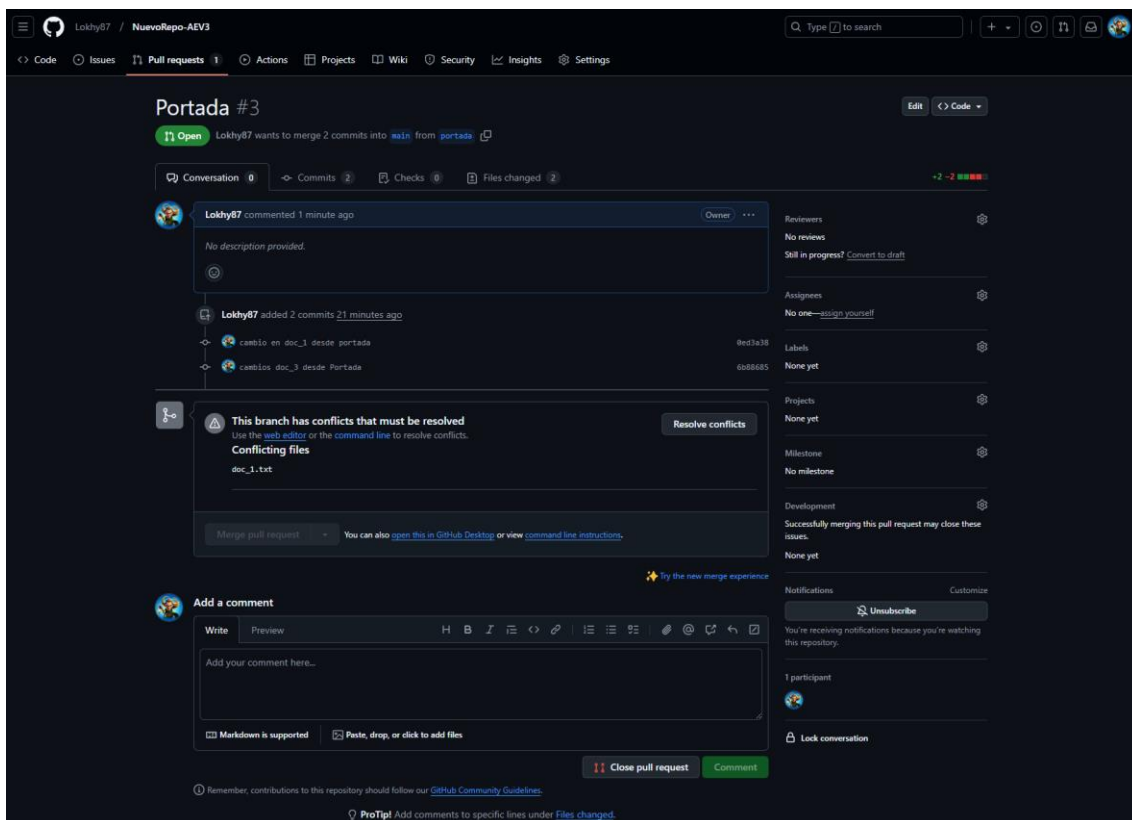
- Realiza alguna modificación en algún fichero del directorio de trabajo (en este caso, mejor si modificas algo que no se haya modificado en las ramas “error” ni “master”). Consolida los cambios.

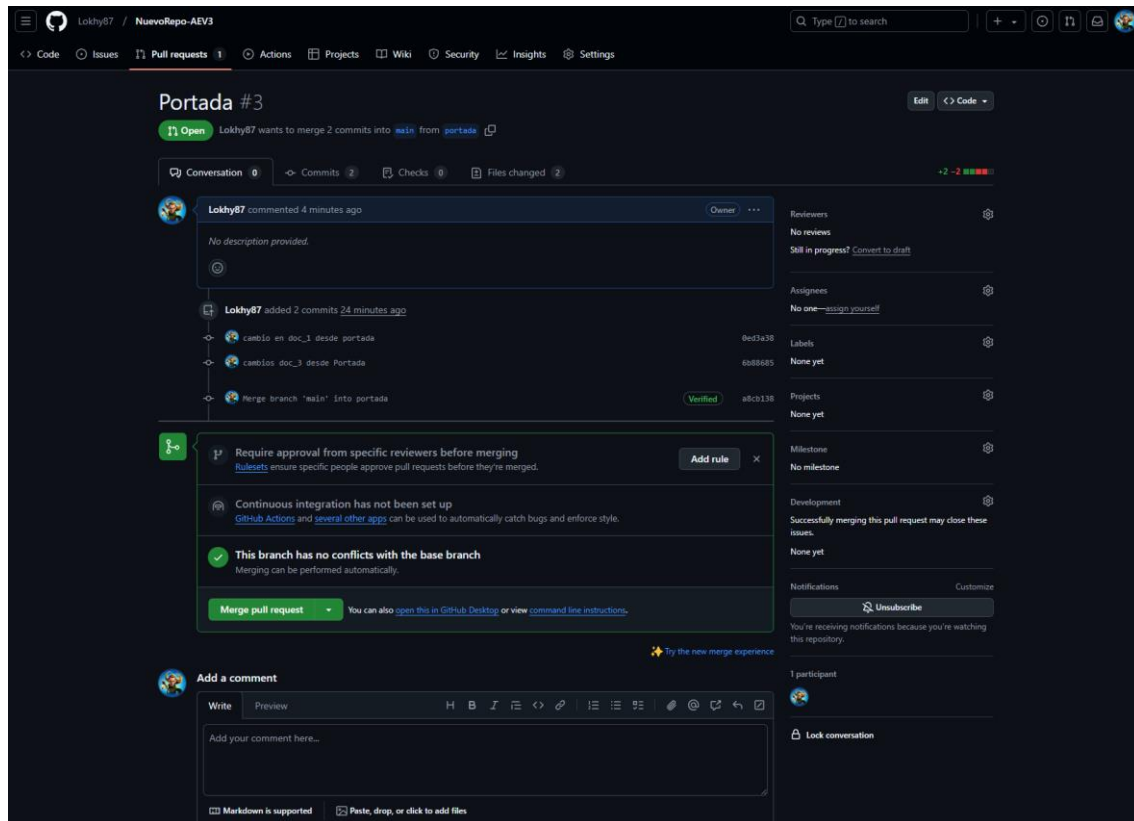


- Integra los cambios en la rama “master” mediante una fusión.

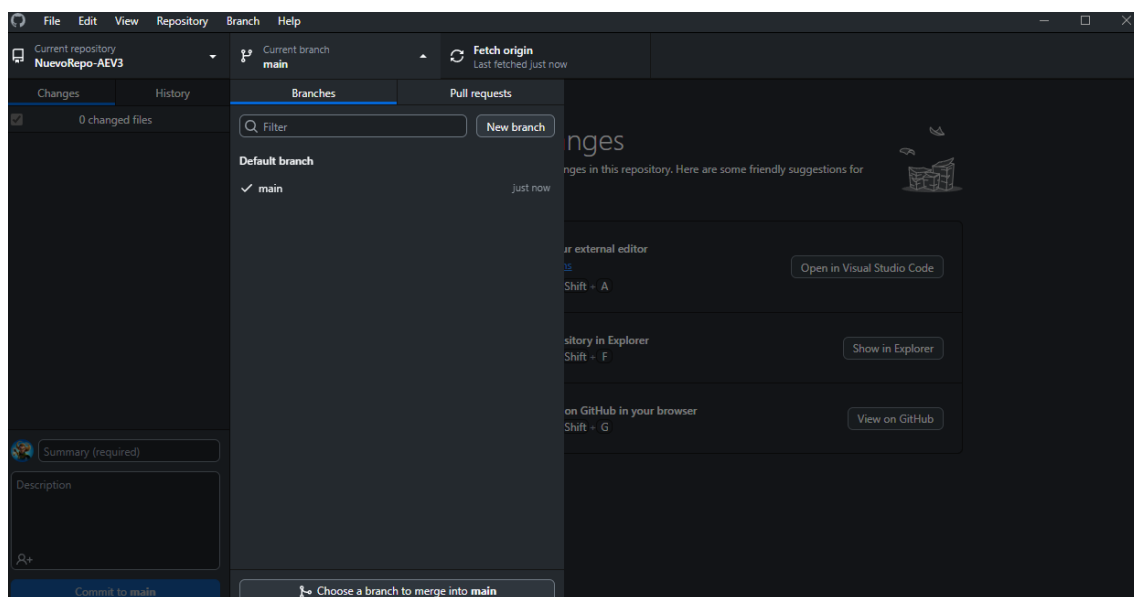


- Explica el tipo de fusión que se haría en este caso. Si ha sucedido algo particular, explícalo con detalle y explica también como lo gestionarías.

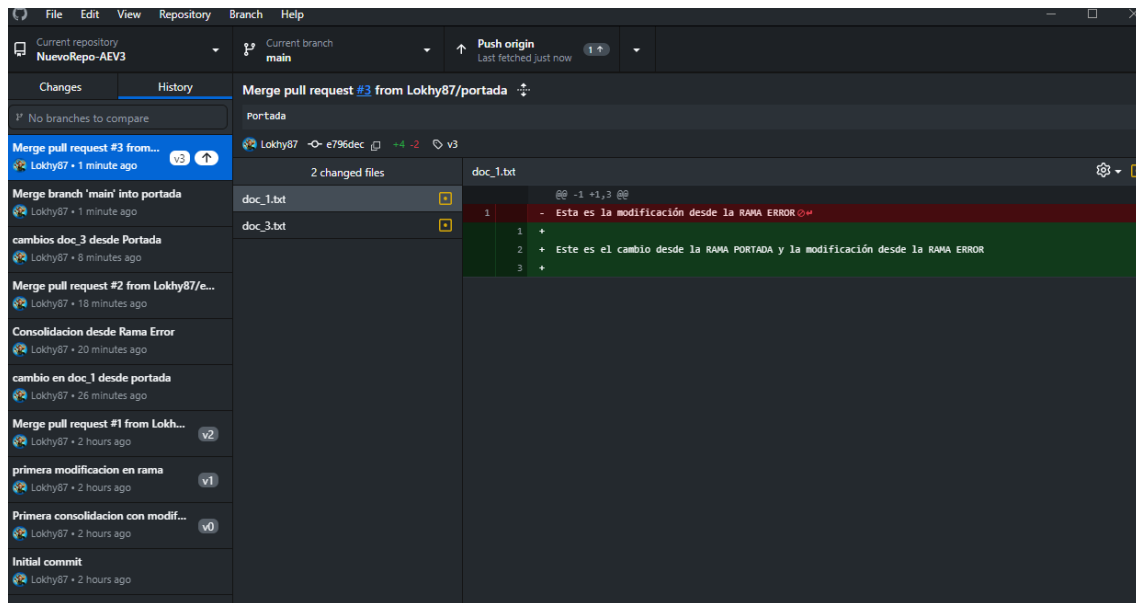




- Llegados a este punto, como hemos terminado la portada y la hemos integrado junto con la corrección del error en la rama “master”, podemos borrar la rama “portada”. Bórrala y continúa los pasos.



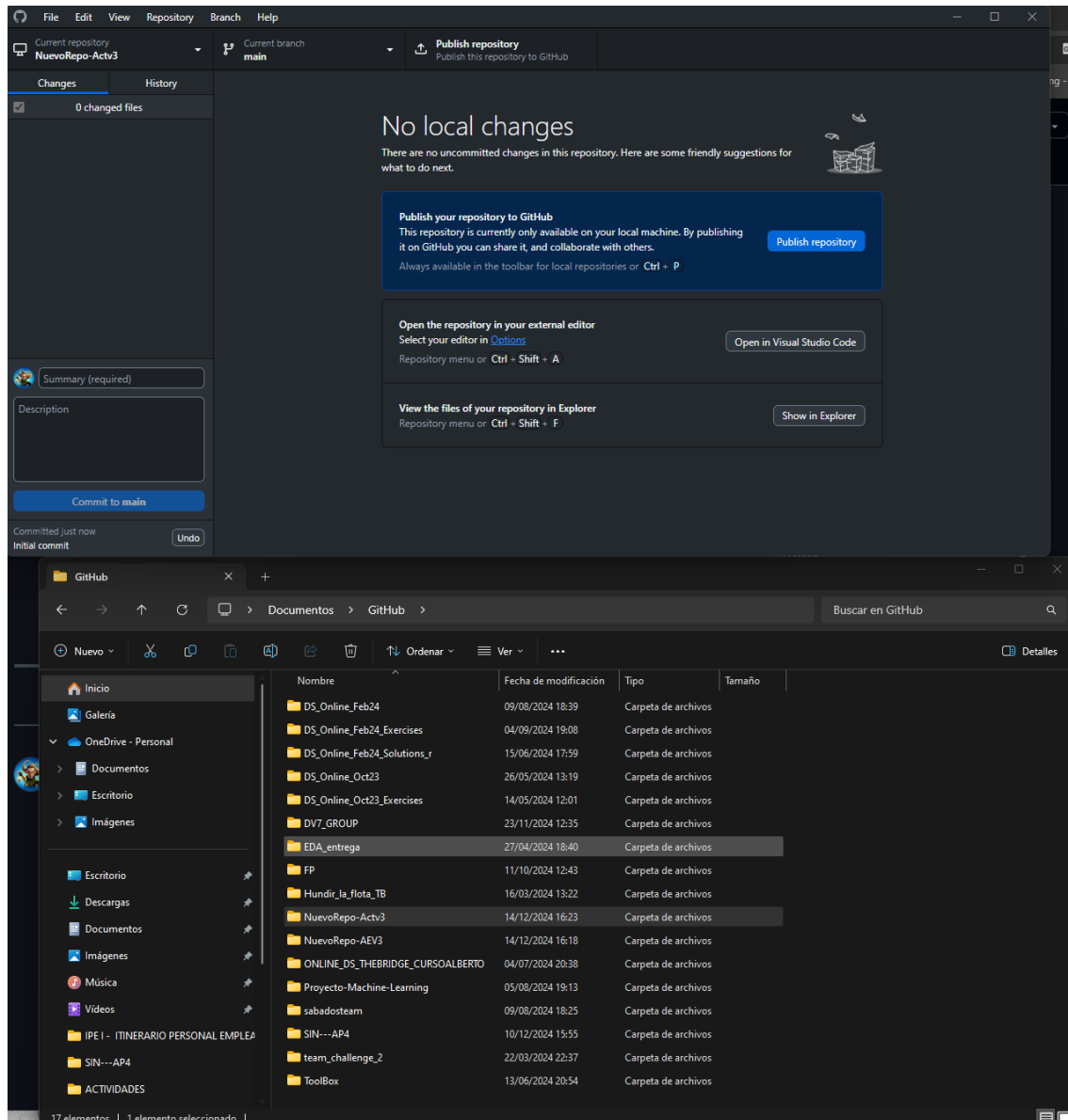
- Etiqueta la última consolidación como v3.



Me saltaba un error, al mergear la rama con la master. Al detectar el error, he modificado el archivo aceptando las 2 indicaciones

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.

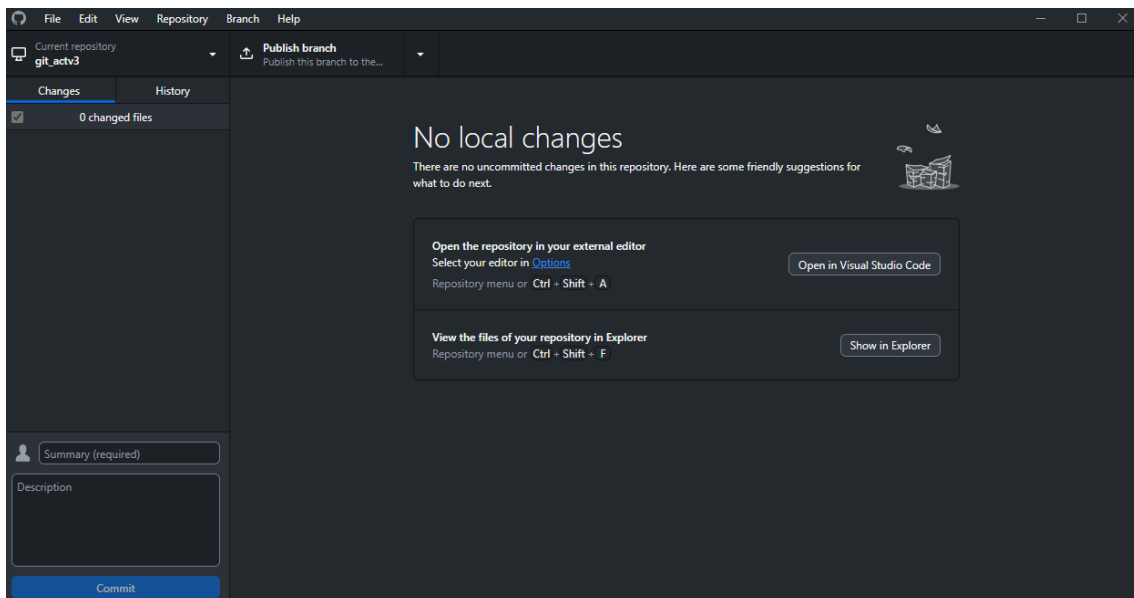
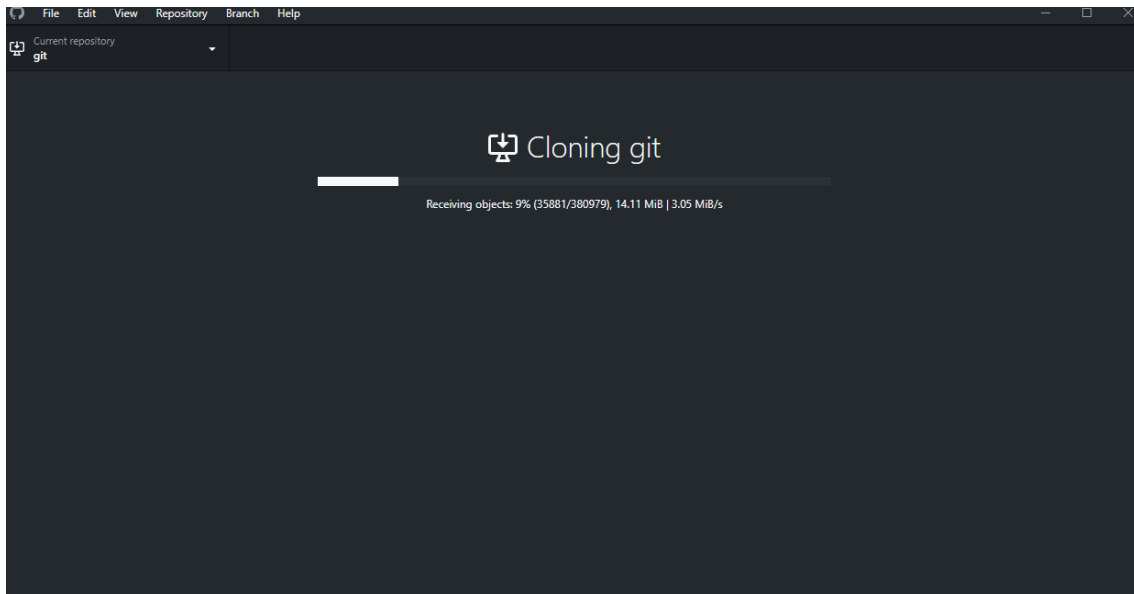
The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a search bar and navigation icons. The main heading is 'Create a new repository' with a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, a note states 'Required fields are marked with an asterisk (*)'. The 'Owner' is set to 'Lokhy67' and the 'Repository name' is 'nuevoRepo_remoto', with a green checkmark indicating it's available. A message says 'Great repository names are short and memorable. Need inspiration? How about [reimagined-chainsaw](#) ?'. The 'Description' field is empty. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked. The 'Add .gitignore' section shows a dropdown for 'gitignore template' set to 'None'. The 'Choose a license' section shows a dropdown for 'License' set to 'None'. A note at the bottom says 'You are creating a public repository in your personal account.' A green 'Create repository' button is at the bottom right.

- 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.

The screenshot shows the GitHub repository page for 'nuevoRepo_remoto'. The repository is public and has 0 stars, 1 watching, and 0 forks. The 'About' section is empty. The 'Releases' section shows 'No releases published' with a link to 'Create a new release'. The 'Packages' section shows 'No packages published' with a link to 'Publish your first package'. The 'Languages' section shows 'HTML' at 100.0%. The 'Suggested workflows' section shows two workflows: 'Jekyll using Docker image' and 'SLSA Generic generator'. The main content area shows a list of commits. The first commit is 'Merge pull request #3 from Lokhy67/portada' by 'Lokhy67' 22 minutes ago. The second commit is 'Initial commit' by 'Lokhy67' 2 hours ago. The third commit is 'Merge branch 'main' into portada' by 'Lokhy67' 22 minutes ago. The fourth commit is 'Primera consolidacion con modificacion de los ficheros' by 'Lokhy67' 2 hours ago. The fifth commit is 'cambios doc_3 desde Portada' by 'Lokhy67' 29 minutes ago. The 'README' section is empty with a link to 'Add a README'.

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 esta 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.
2. Modificación archivos locales.
3. Commit de los cambios.
4. sincronización remota.
5. Gestión conflictos (si es necesario)