

Crea tu primer repositorio Git

Este ejercicio lo guiará a través de los primeros pasos para aplicar el control de versiones usando Git a una carpeta y su contenido. Cubrirás:

- habilitar Git en una carpeta
- revisar el estado de los archivos en la carpeta
- edite archivos repetidamente y agregue sus cambios a Git
- revisar los cambios
- comprobar una versión anterior de sus cambios
- aplicar una etiqueta a una versión específica para futuras referencias
- excluir archivos del control de versiones

Requisito previo: SourceTree instalado

Hay muchos clientes para Git, en estos ejercicios usará SourceTree para administrar Git. SourceTree es un cliente Git para Windows y Mac. Si aún no lo ha instalado, hágalo siguiendo estas instrucciones: https://nerc-ceh.github.io/version_control/install_sourcetree_windows

Nota sobre la línea de comandos

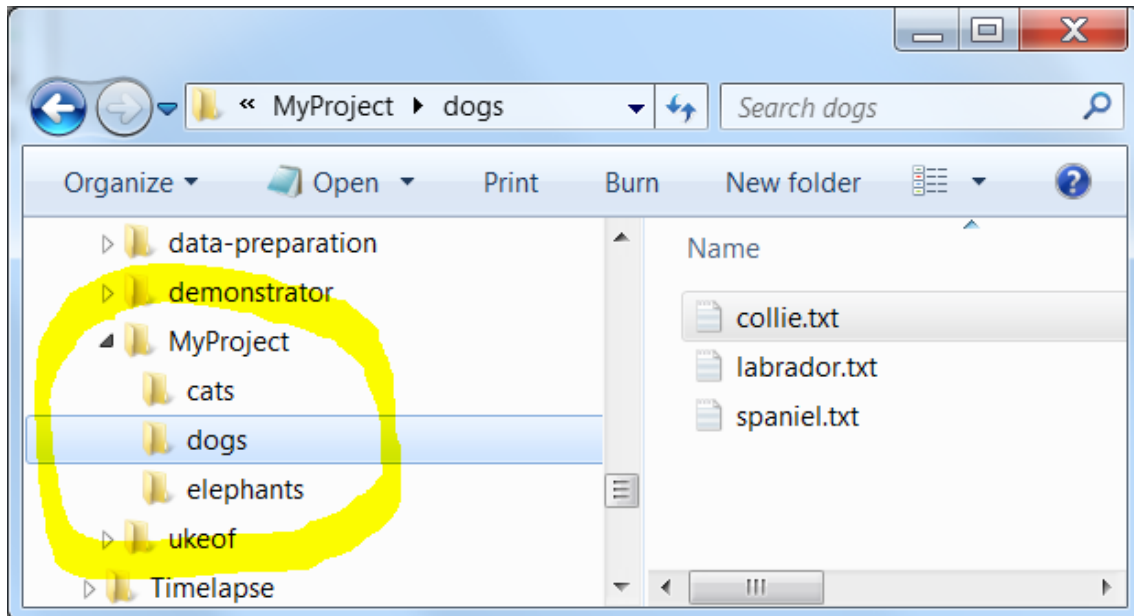
SourceTree le brinda una interfaz gráfica para administrar Git y nuestros ejercicios usan esa interfaz. Sin embargo, Git se puede usar desde la línea de comandos y, en última instancia, esta puede ser su opción preferida. Para su referencia, hemos incluido este script https://nerc-ceh.github.io/version_control/commandline_exercise1.txt, que le muestra cómo hacer el ejercicio 1 completamente desde la línea de comandos. Puede ingresar comandos en SourceTree si va a: Acciones > Abrir en Terminal.

Sin embargo, para los ejercicios de clase, utilice Source Tree como se muestra a continuación en lugar de intentar la línea de comando, ya que esto hace que sea más fácil para los capacitadores.

Paso 1. Crea tu proyecto de trabajo

Este paso crea una carpeta de proyecto con subcarpetas y archivos listos para el resto del ejercicio.

- Descargue https://nerc-ceh.github.io/version_control/MyProject.zip y descomprima en una ubicación de trabajo conveniente (por ejemplo, en su carpeta **Documentos**). Esto le proporciona una carpeta de trabajo que contiene subcarpetas y archivos para editar y controlar la versión. Deberías tener algo como esto:

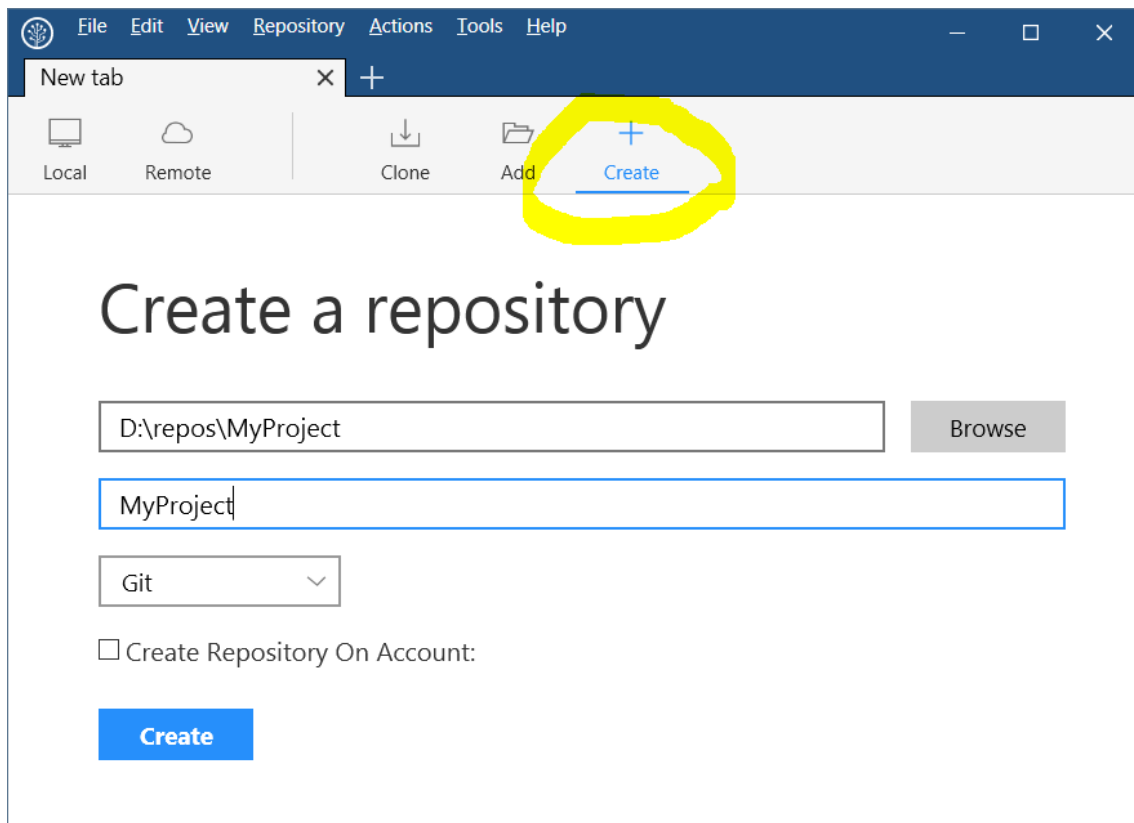


- La ubicación de la carpeta raíz (por ejemplo, C:\Users\jon\Documents\MyProject) será necesaria en el Paso 2.

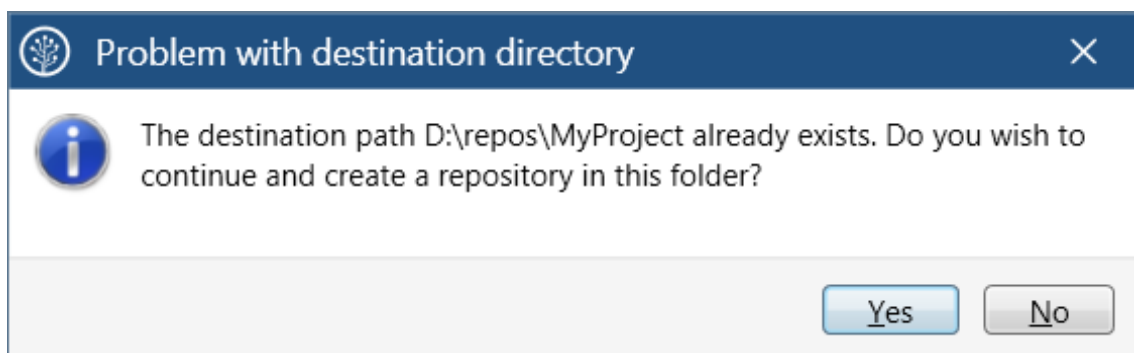
Paso 2. Habilite Git en su nuevo proyecto

Este paso agrega el repositorio de Git a la raíz de la carpeta de su proyecto.

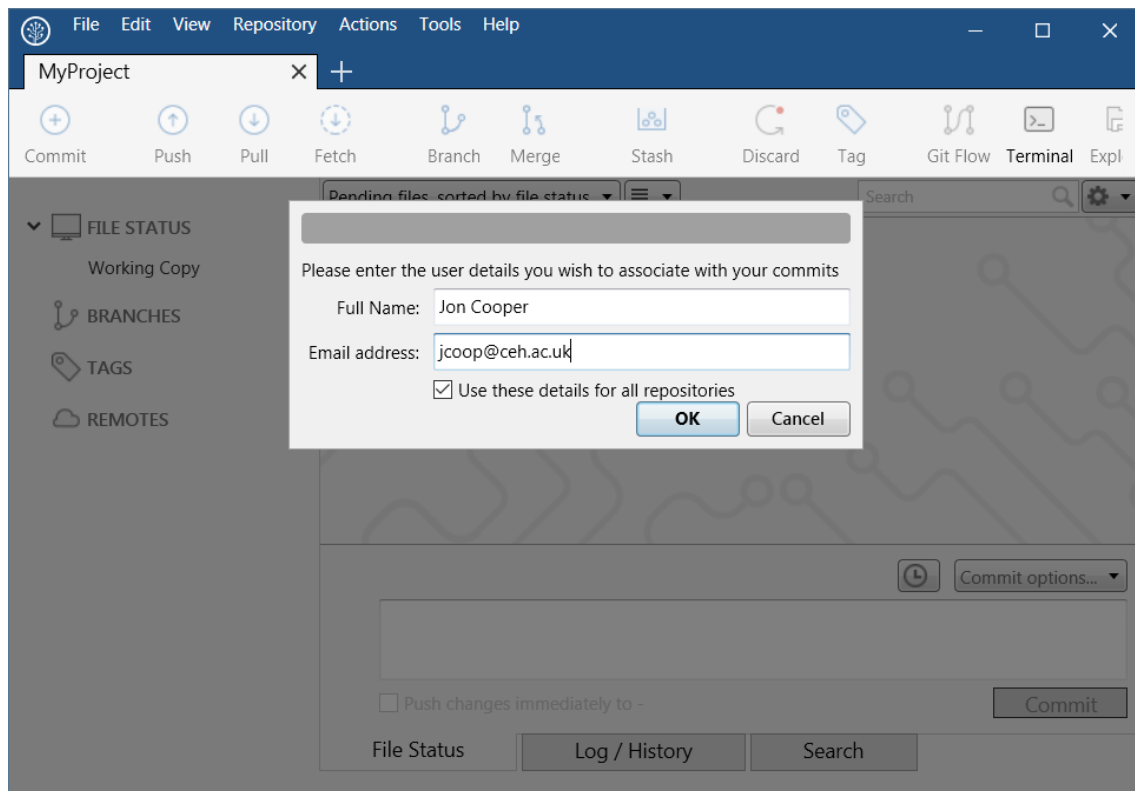
- Árbol de código abierto
- Presione el botón **Crear** para mostrar el formulario **Crear un repositorio** .
- Rellene los campos de la siguiente manera:
 - **Ruta de destino** : la ubicación de la carpeta raíz del paso 1; en otras palabras, la carpeta que desea controlar (por ejemplo, C:\Users\jon\Documents\MyProject)
 - **Nombre** : un nombre simple para identificar su repositorio en SourceTree
 - Desplegable **de Git o Mercurial** : **seleccione Git**
 - **Crear repositorio en el menú** desplegable de la cuenta: asegúrese de que **no** esté seleccionado



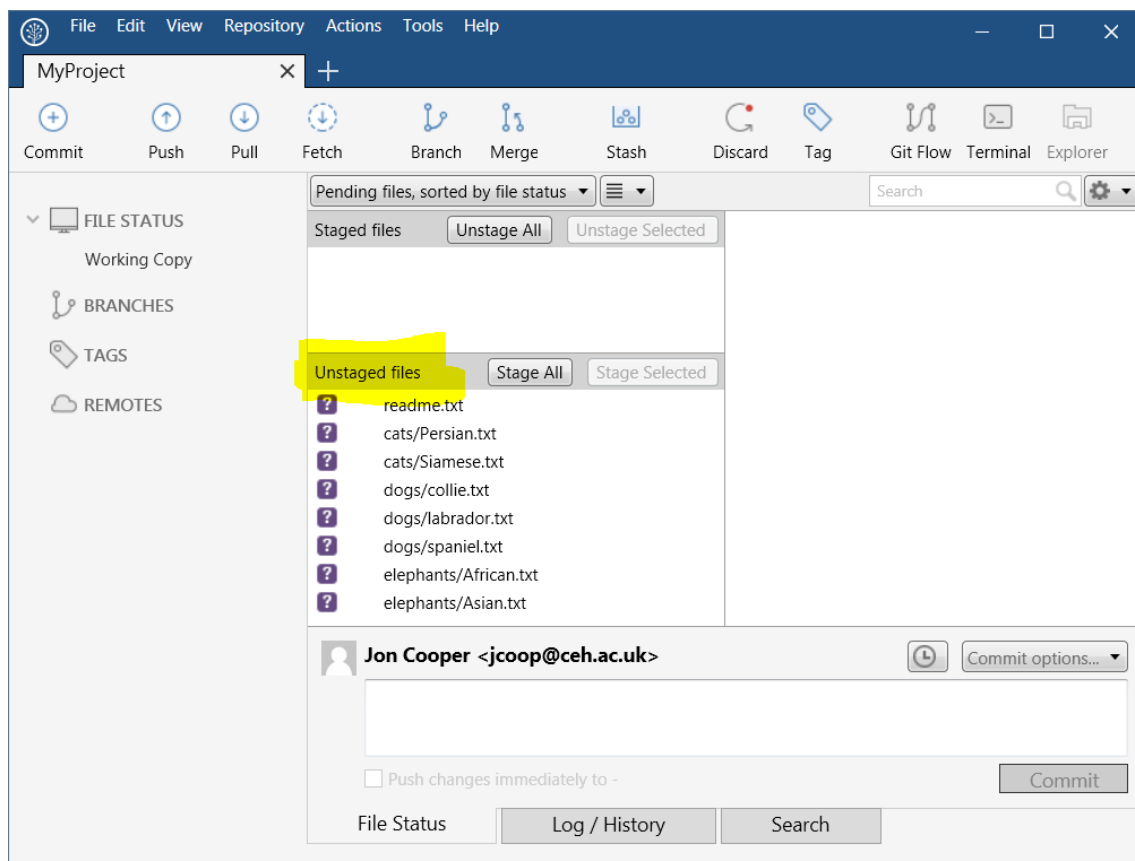
- Presiona el botón **Crear** , para crear un repositorio Git en la carpeta raíz de tu proyecto
- Debería ver la siguiente advertencia, que simplemente le alerta sobre el hecho de que agregará un repositorio de Git a una carpeta existente, en lugar de crear la carpeta por usted (que es otra forma de iniciar su proyecto de versión controlada): presione sí **para** aceptar :



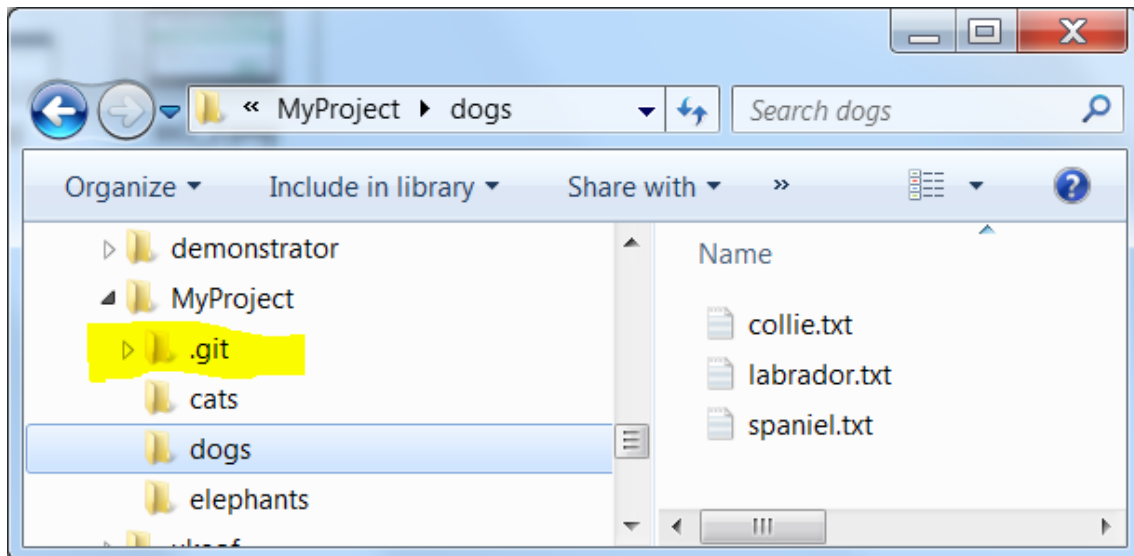
- **Es posible** que se le solicite que proporcione su nombre y correo electrónico. Si ha configurado esto antes, entonces no le preguntará. Si es necesario, ingrese sus detalles y marque **Usar estos detalles en todos los repositorios** para que no tenga que volver a hacerlo. Sus datos se utilizan para agregar su identidad al registro cuando realiza **ediciones** :



- SourceTree ahora mostrará el estado de su proyecto de versión controlada, con todos los archivos en el área **de archivos sin preparar** :



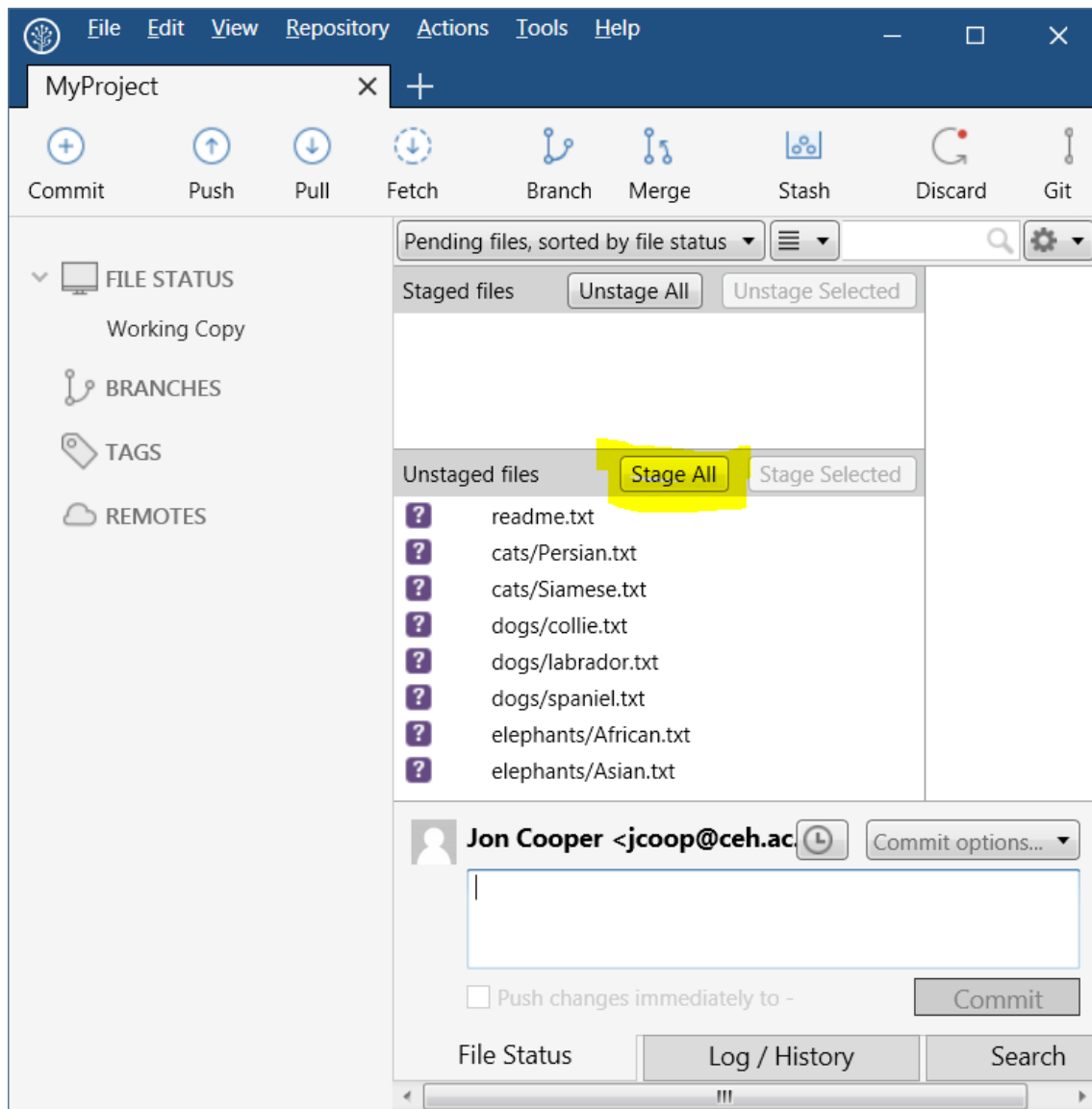
- Además, si busca en la raíz de su proyecto en el Explorador de Windows, ahora verá una nueva carpeta llamada '.git', este es el repositorio de Git (es posible que deba habilitar 'Mostrar carpetas ocultas' para verlo)



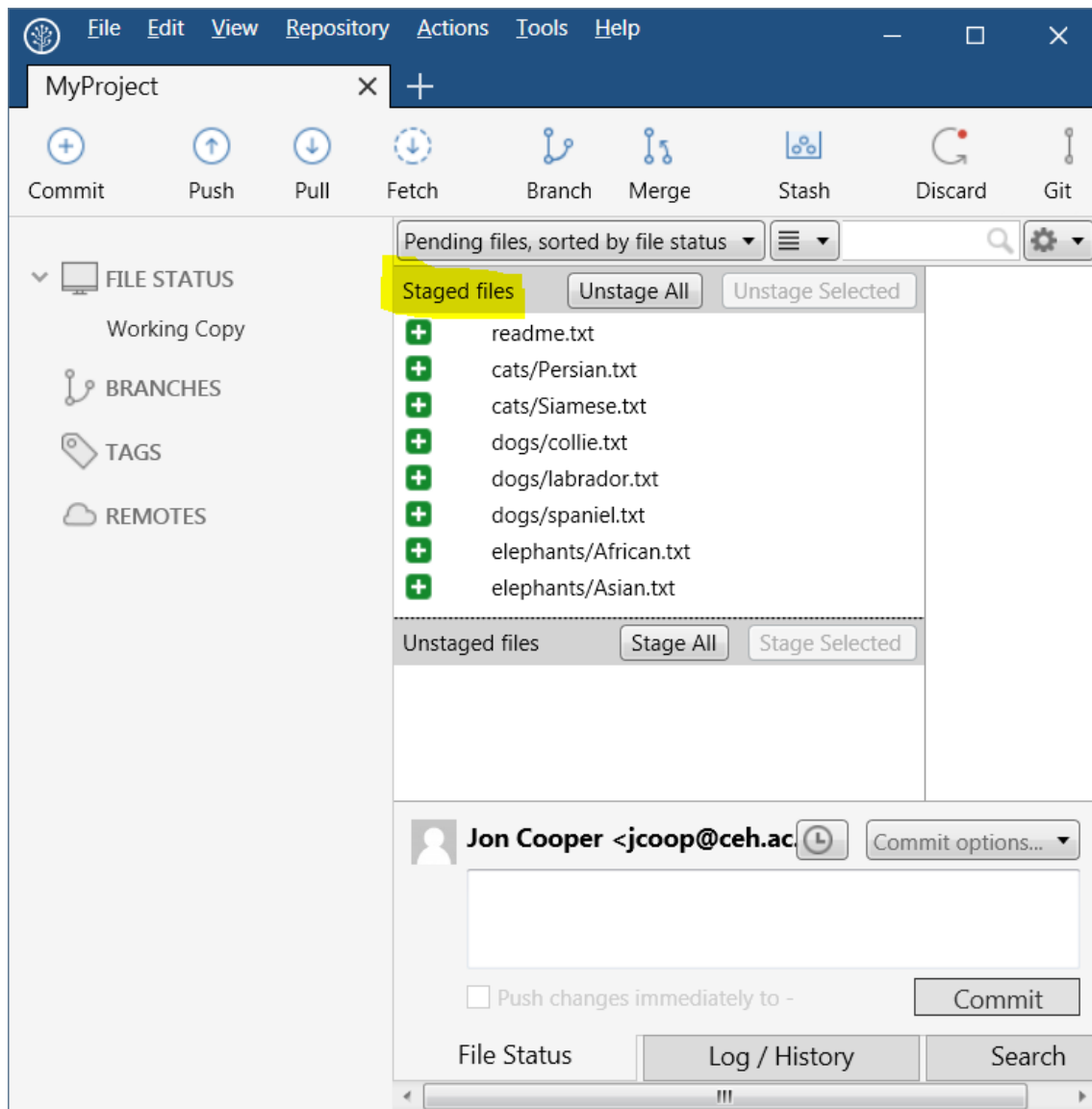
Paso 3. Complete su nuevo repositorio

Su nuevo repositorio de Git aún no contiene ningún archivo para el control de versiones, es decir, todos los archivos están actualmente **sin seguimiento** . Es por eso que tienen un signo de interrogación junto a ellos en SourceTree. Este paso coloca todos sus archivos en Git, que es un proceso de dos pasos: **prepararlos** y luego **confirmarlos** .

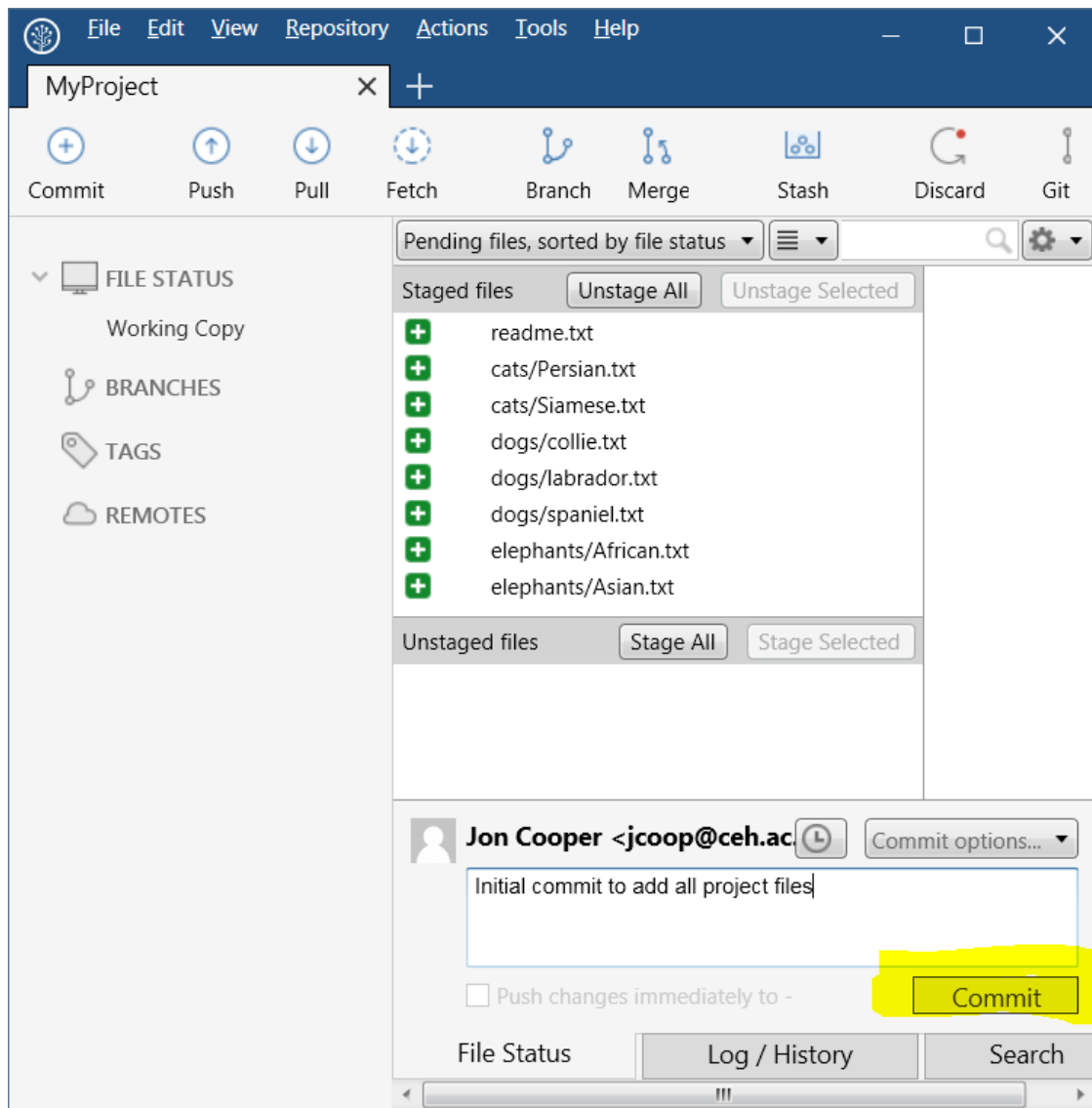
- Agregue todos los archivos al área **de archivos preconfigurados** presionando el botón **Preparar todo** .



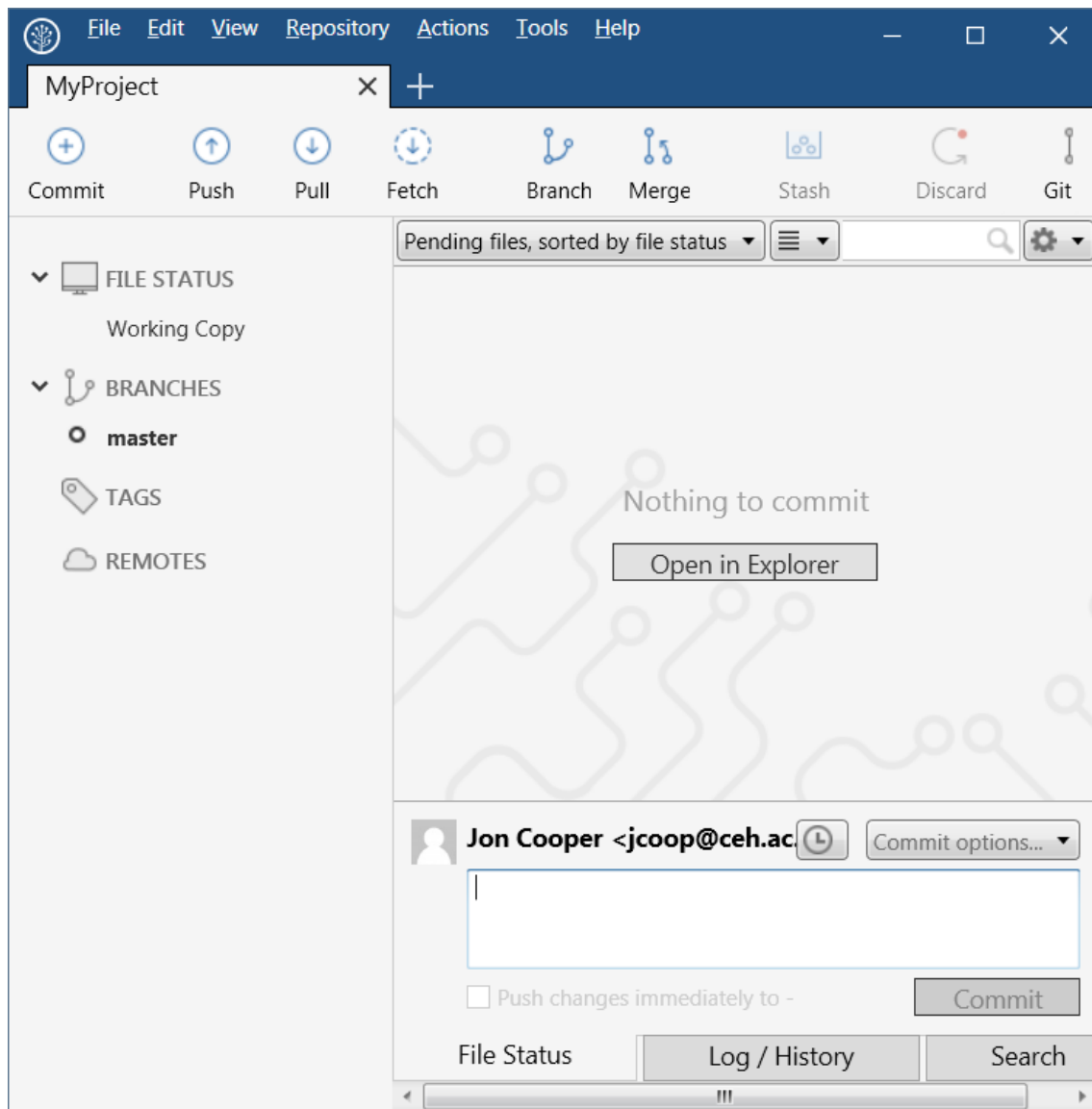
- Tenga en cuenta que ahora se muestran en el área **de archivos preconfigurados** con un ícono verde más, lo que indica que son archivos completamente nuevos listos para enviarse al repositorio de Git.



- Escriba un mensaje de compromiso, por ejemplo, 'Compromiso inicial para agregar todos los archivos del proyecto' y presione **Confirmar**



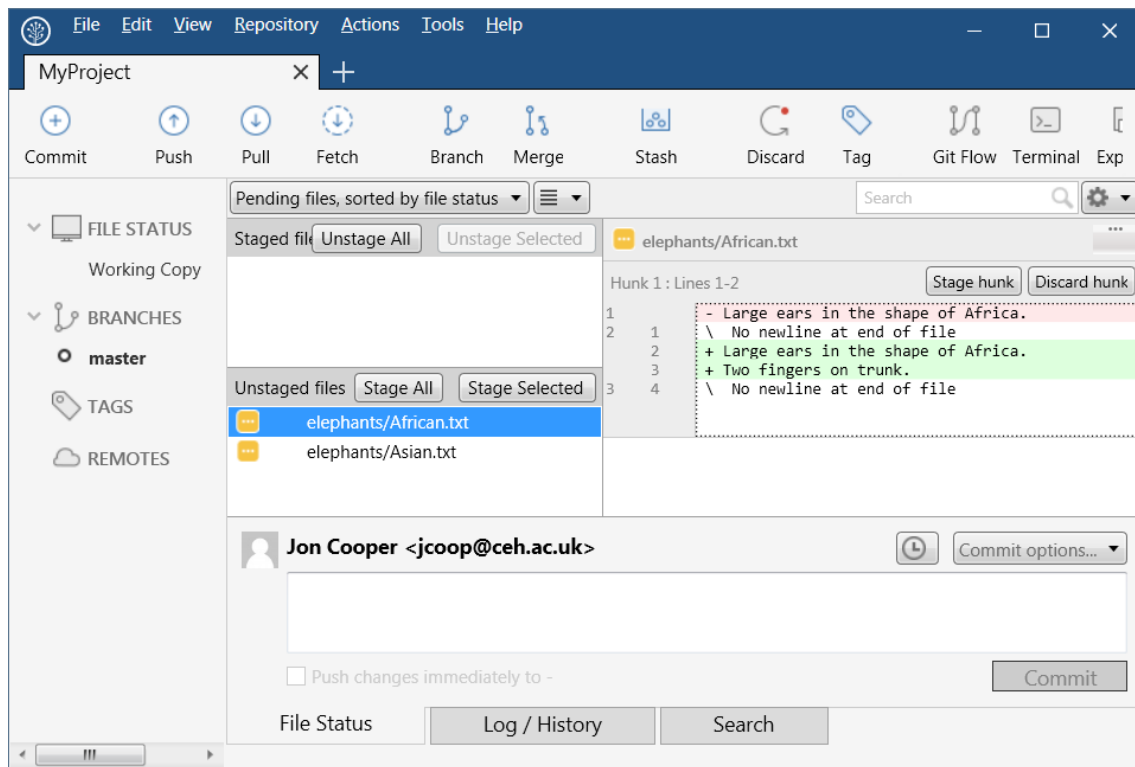
- Todos los archivos están comprometidos con el repositorio de Git y las áreas de 'Archivos no preparados' y 'Archivos preparados' están vacías



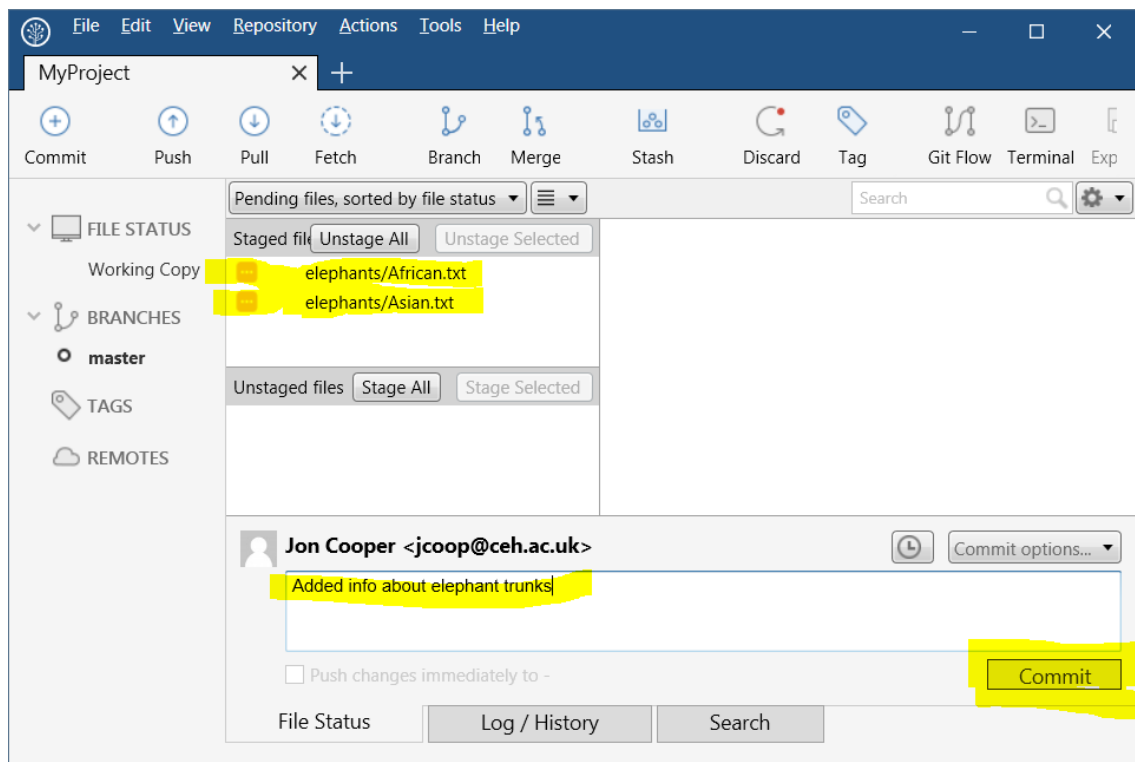
Paso 4. Edite archivos y confirme cambios en Git

En este paso, editará algunos archivos y enviará sus cambios a Git.

- En su editor de texto favorito, abra un par de archivos descargados (p. ej., elephants/Asian.txt y elephants/African.txt) y edítelos.
- En SourceTree, los archivos que editó se muestran en el área **Sin organizar** con un ícono amarillo '...' junto a ellos. Si resalta uno, se muestran sus cambios:



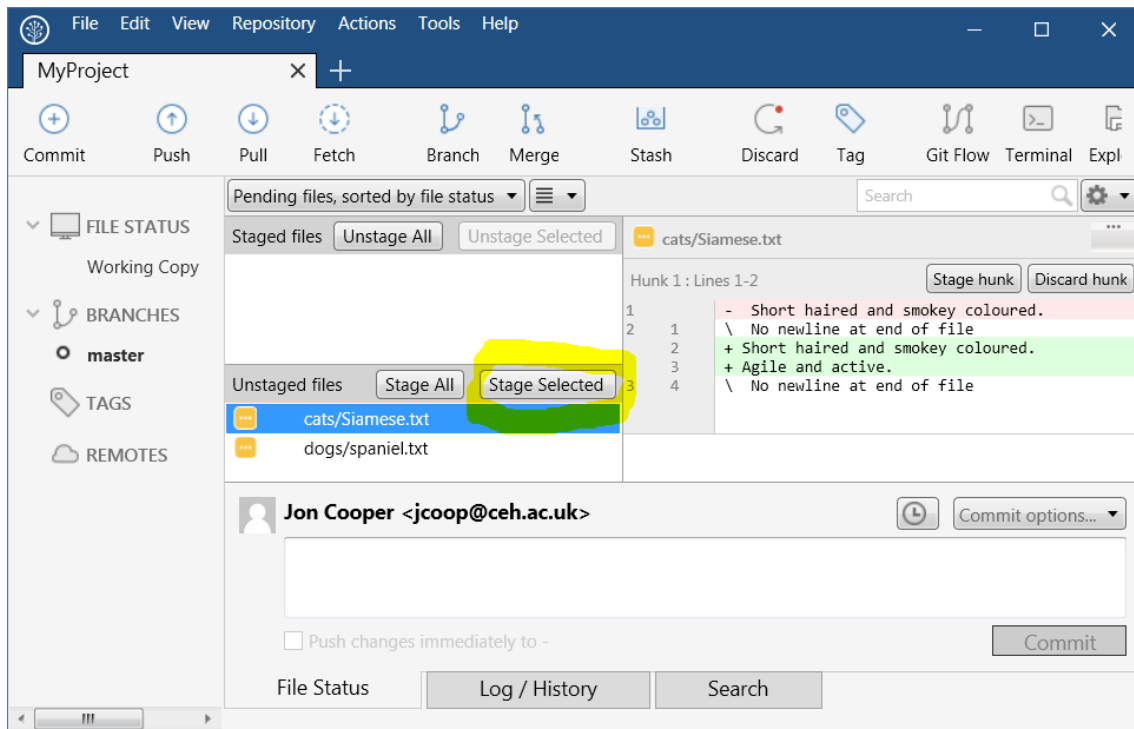
- Agregue al área de preparación y confirme con un mensaje adecuado como lo hizo antes.



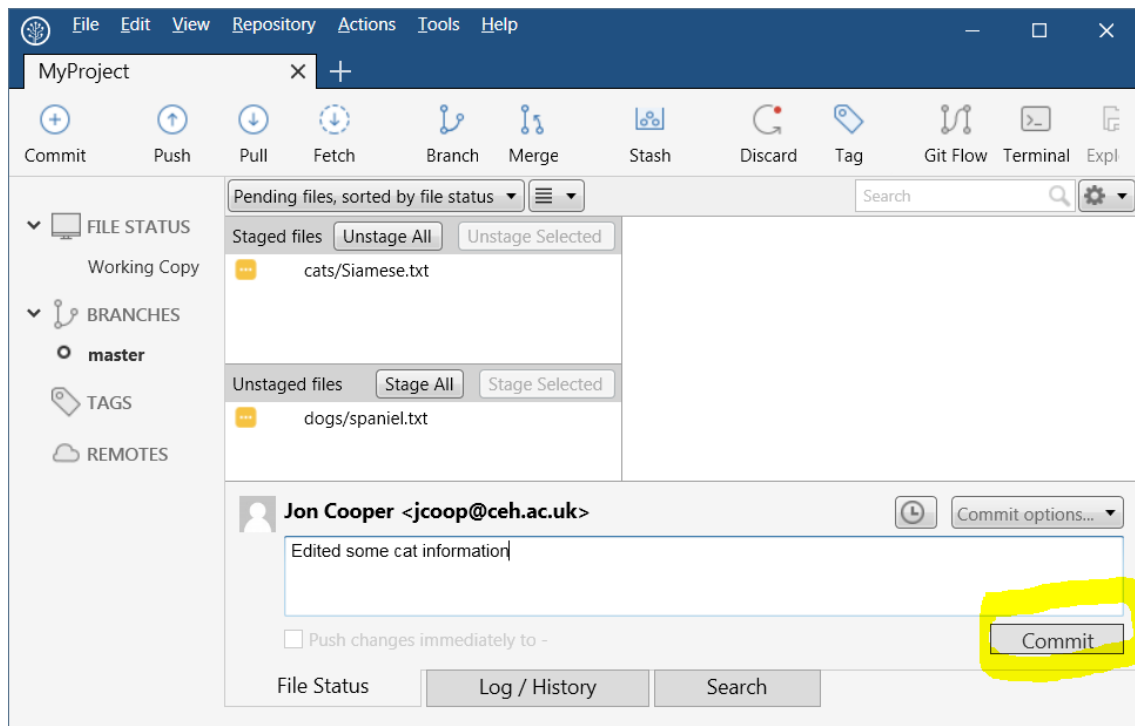
Paso 5. Archivos específicos de etapa

En este paso, editará dos archivos, pero los **preparará** y **confirmará** por separado para ilustrar cómo se pueden usar las confirmaciones para organizar sus ediciones.

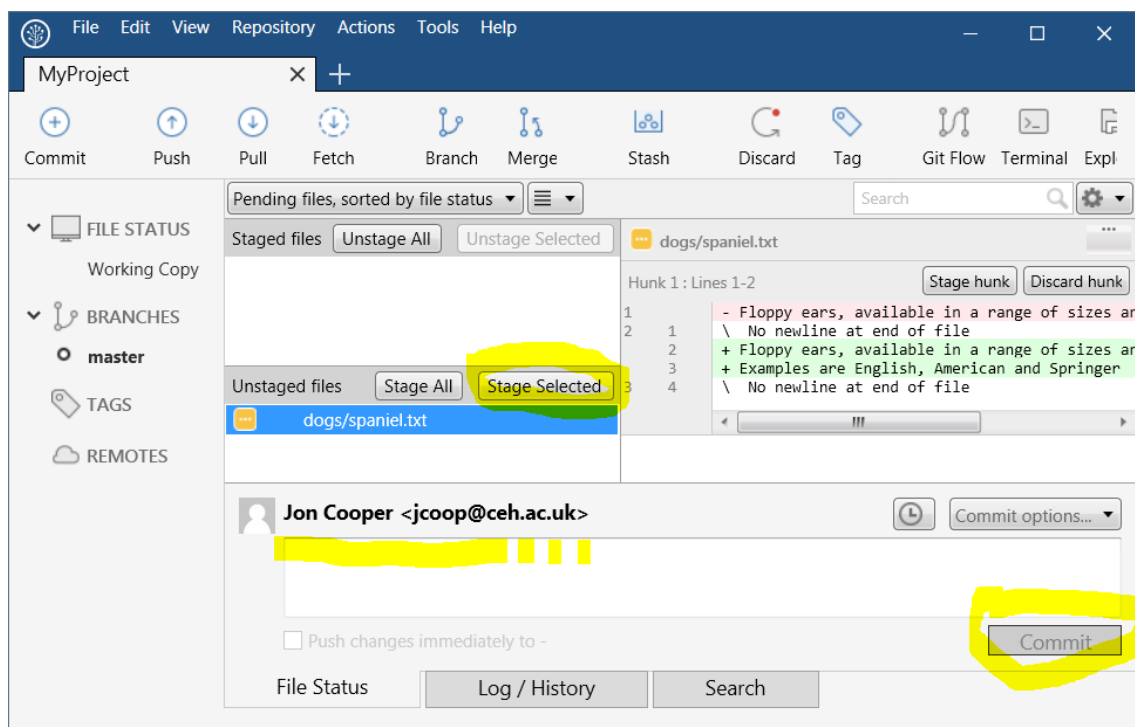
- Agregue texto nuevo a **dos** archivos (por ejemplo, cats/Siamese.txt y dogs/spaniel.txt)
- En **Archivos sin etapas**, seleccione solo **un** archivo editado, luego presione **Etapas seleccionadas**



- Esta vez, solo el archivo seleccionado está listo para confirmar. Dale un mensaje apropiado y luego **Confirmar**



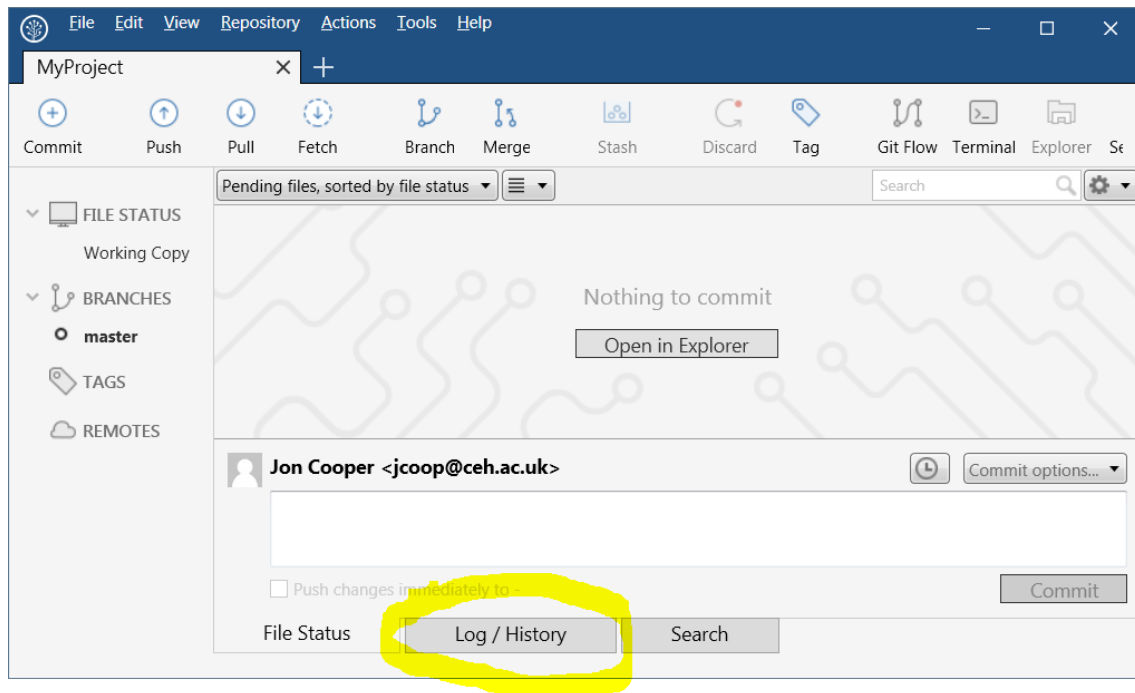
- Verá que queda un archivo en **Archivos sin preparar** , póngalo en escena y confirme con un mensaje:



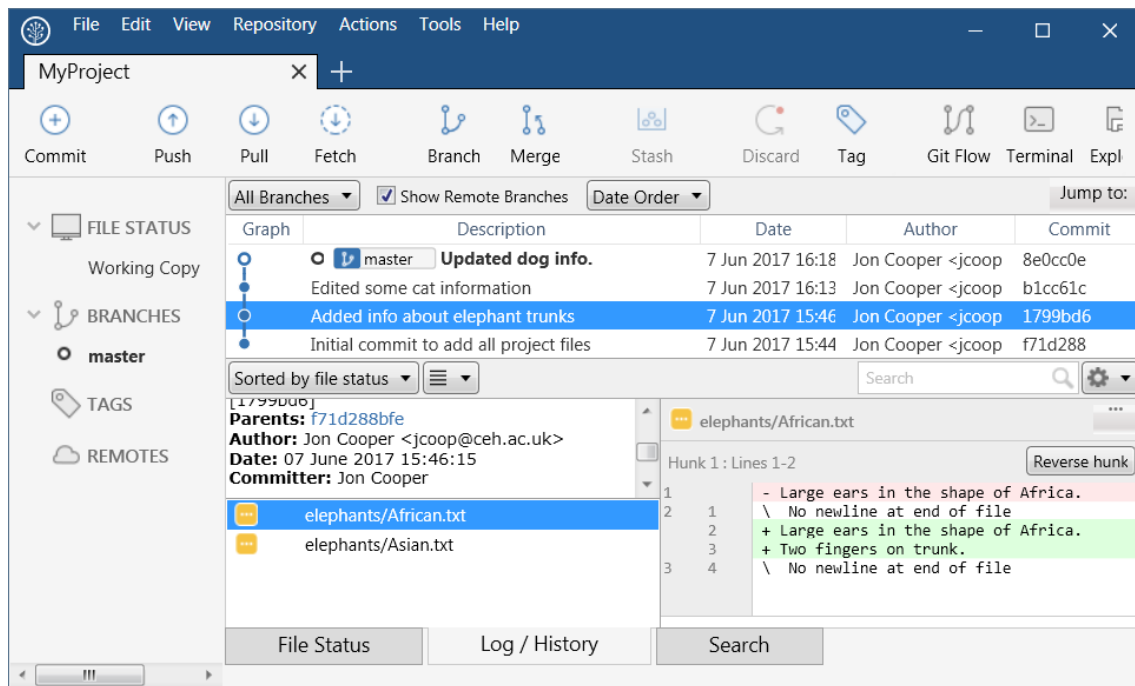
Paso 6. Revise el registro de Git y obtenga una versión

Ahora debería tener un rango de cambios de archivos confirmados en Git. En este paso, verá el registro de sus ediciones y verificará una de esas revisiones.

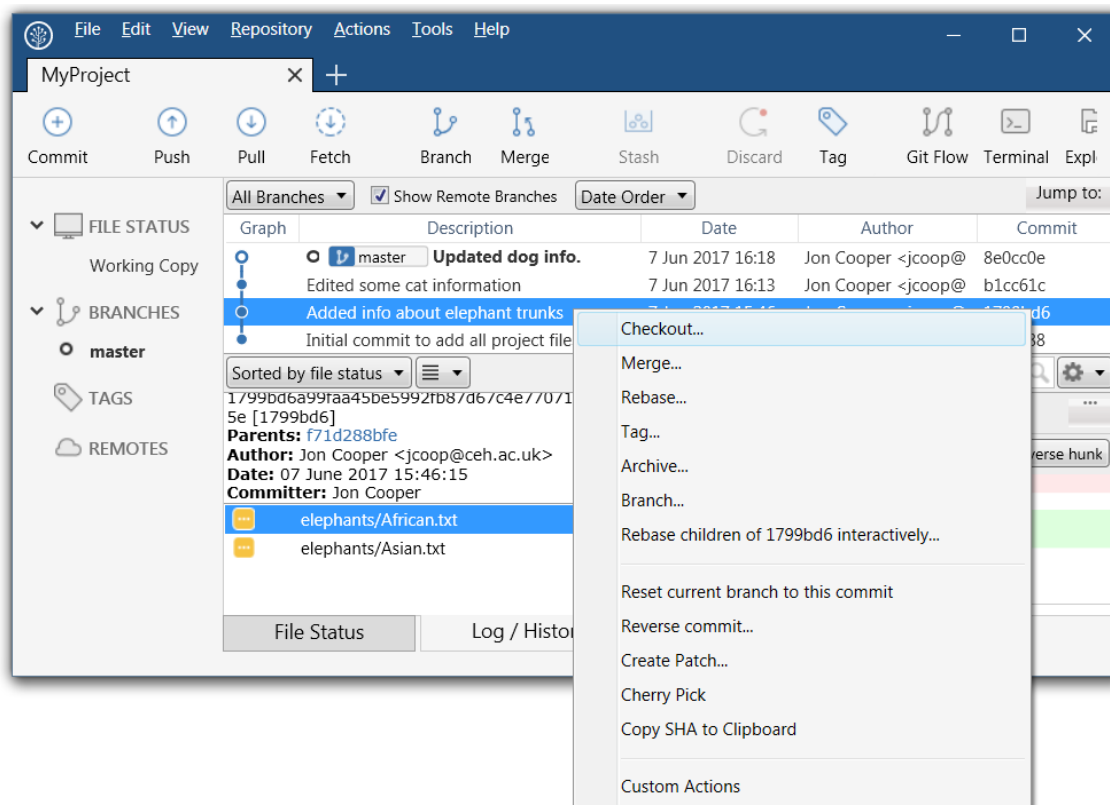
- Eche un vistazo al registro de Git presionando la pestaña **Registro/Historial** en la parte inferior:



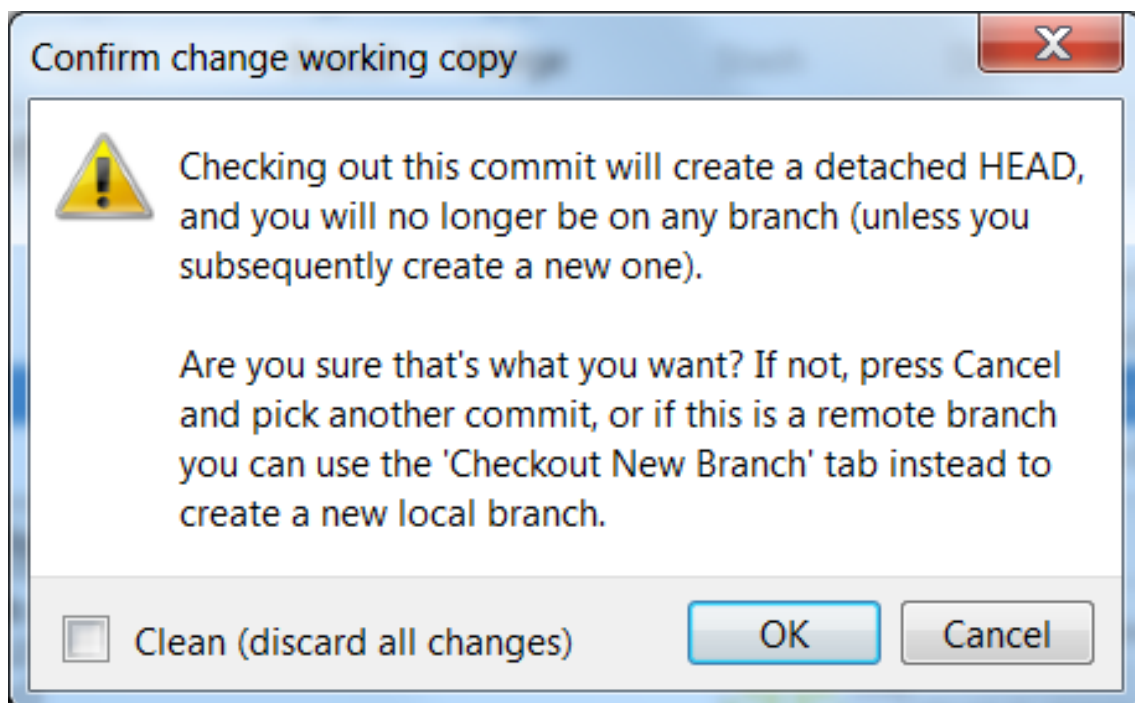
- Hay 3 paneles que muestran diferentes aspectos del registro:
 - i. en primer lugar, cada confirmación junto con su mensaje de confirmación se muestra en el panel superior.
 - ii. seleccione una confirmación en el panel superior para mostrar qué archivos se editaron (panel inferior izquierdo)
 - iii. seleccione un archivo en el panel inferior izquierdo para mostrar sus ediciones (panel inferior derecho)



- ahora verificará una de las confirmaciones que se muestran en el registro. Cuando realiza el pago de una confirmación específica, todos los archivos en su copia de trabajo serán exactamente como estaban cuando realizó esos cambios. En su registro, haga clic con el botón derecho en una confirmación anterior en el panel superior (p. ej., en mi ejemplo 'Información agregada sobre trompas de elefante') y seleccione **Pagar...** en el menú contextual:

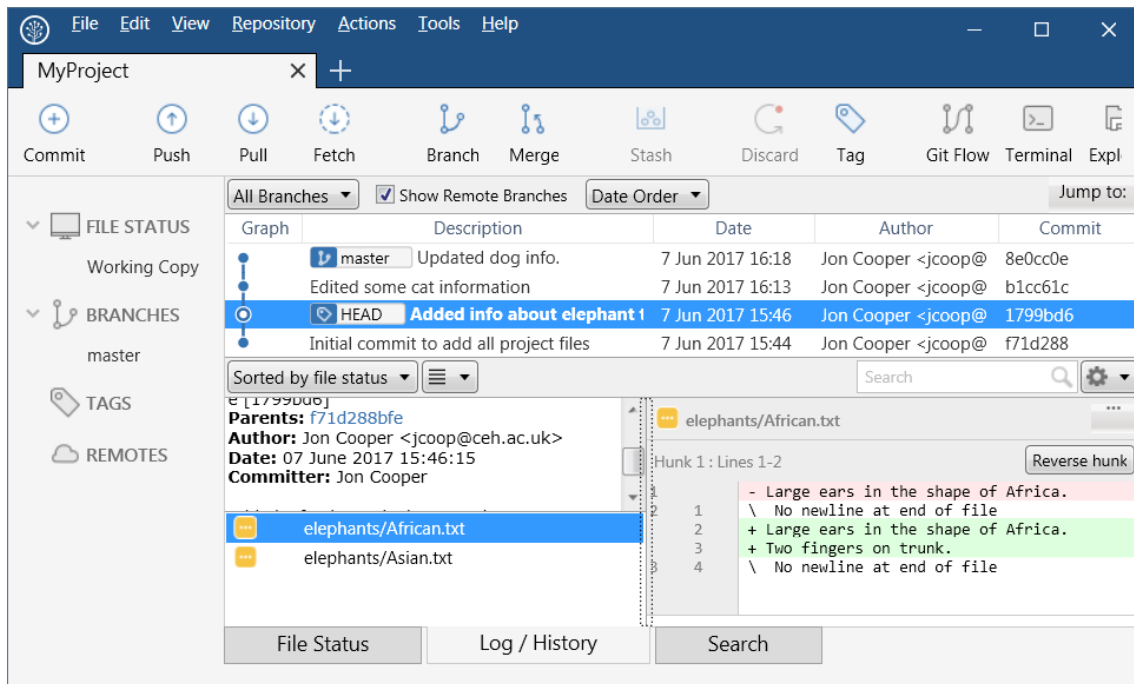


- verá la siguiente advertencia, para este ejercicio es seguro ignorarla (NO marque **Limpiar (descartar todos los cambios)**):



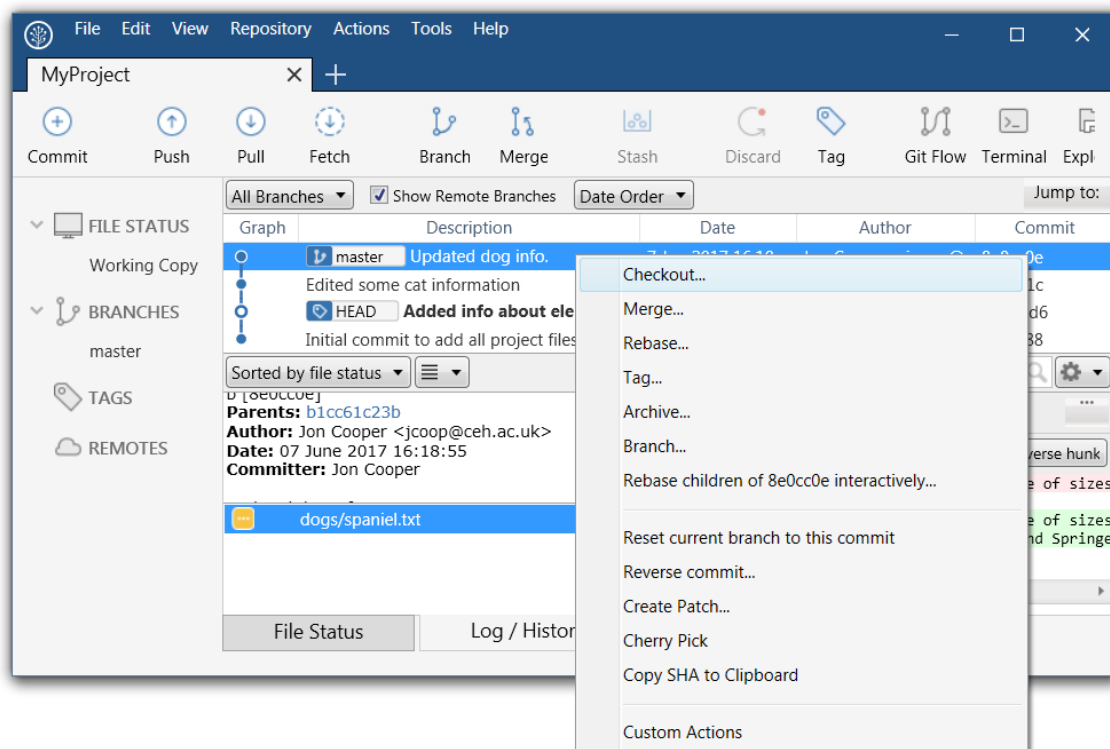
- el registro muestra la etiqueta **HEAD** en la confirmación que acaba de verificar; esto indica que todos los archivos en su copia de trabajo ahora están como estaban cuando realizó esa confirmación. Si observa los

archivos en esa carpeta en el Explorador de Windows, verá todo el trabajo realizado después de que falte esa confirmación. Esta es una forma de obtener versiones anteriores de archivos.

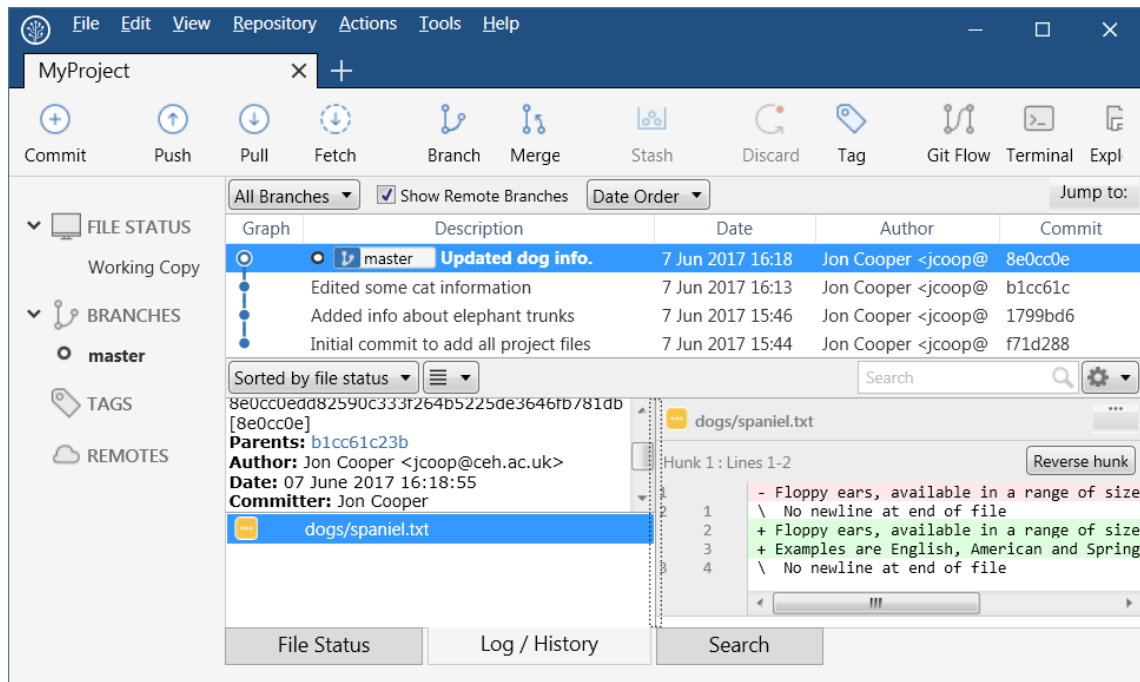


S

- Vuelva a la última versión de su trabajo revisando la última confirmación de la rama **maestra** ; esto se hace haciendo clic con el botón derecho en la **maestra** y seleccionando **Pagar...** en el menú contextual.



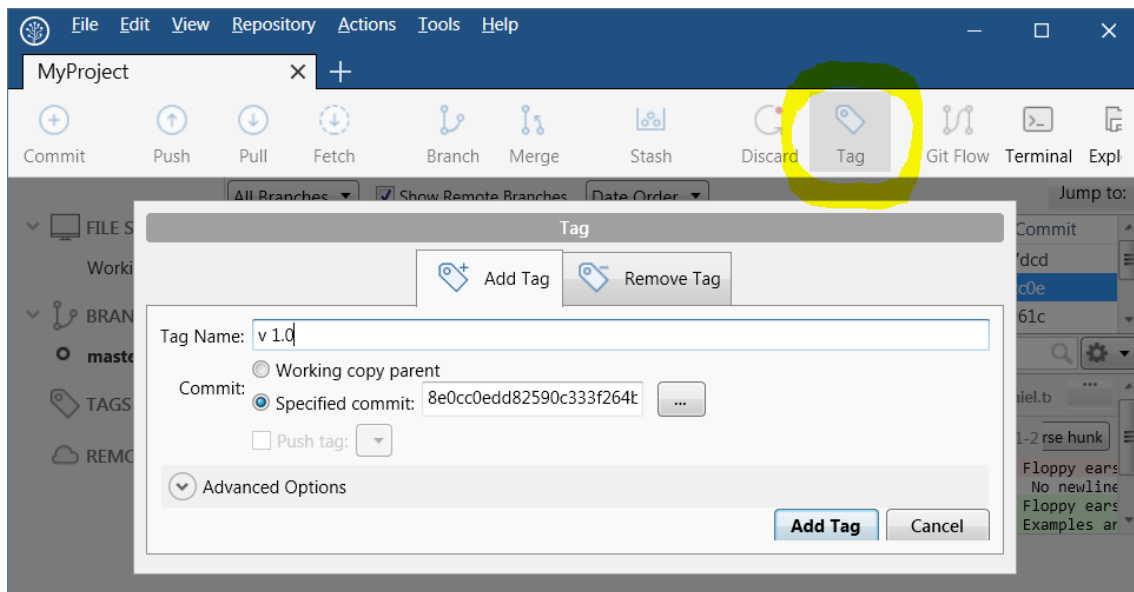
- Ahora debería volver a su última edición, con los archivos en su carpeta de trabajo que contienen todas las ediciones.



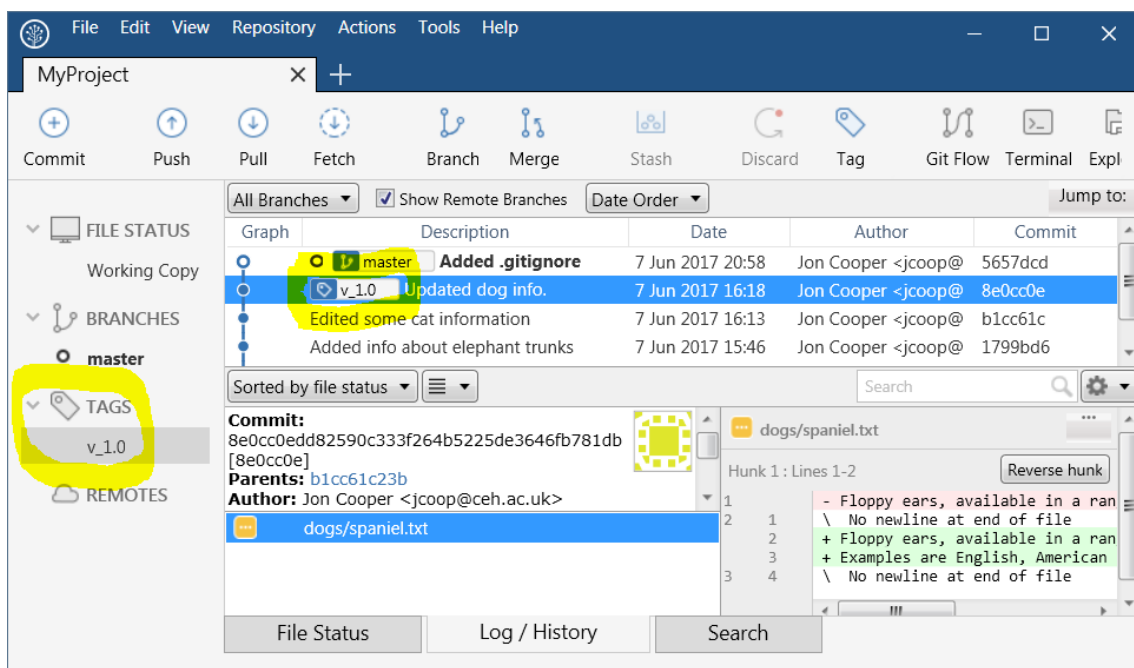
Paso 7. Etiquetado

Como puede imaginar, el registro de Git pronto se llena con muchas confirmaciones. Hay momentos en los que querrá identificar un punto específico en el registro como algo más significativo, como cuando haya completado grandes bloques de trabajo, alcanzado un hito o finalizado un cambio importante en el modelado o el código del sitio web. Este es un punto de publicación al que puede referirse a través de un número de versión, por ejemplo, v1.0.1 de mi modelo o sitio web. En Git esto se llama una **etiqueta anotada**. El estado completo de su repositorio se almacena en esta etiqueta. Este paso agregará una etiqueta anotada a su repositorio.

- Seleccione la opción **Etiqueta** resaltada a continuación, complete el diálogo de esta manera:
 - **Nombre de la etiqueta** : su número de versión (por ejemplo, v1.0.0)
 - **Copia principal de trabajo o confirmación especificada** : elija el estado actual de su repositorio o una confirmación específica a la que se vinculará esta etiqueta
 - **Opción avanzada** : para este ejercicio no los estamos usando
 - **Agregar etiqueta** : presione para agregar la etiqueta



- Ahora verá la etiqueta en su vista **de Registro/Historial** y también puede enumerar todas las etiquetas a la izquierda seleccionando el menú desplegable **Etiquetas** :

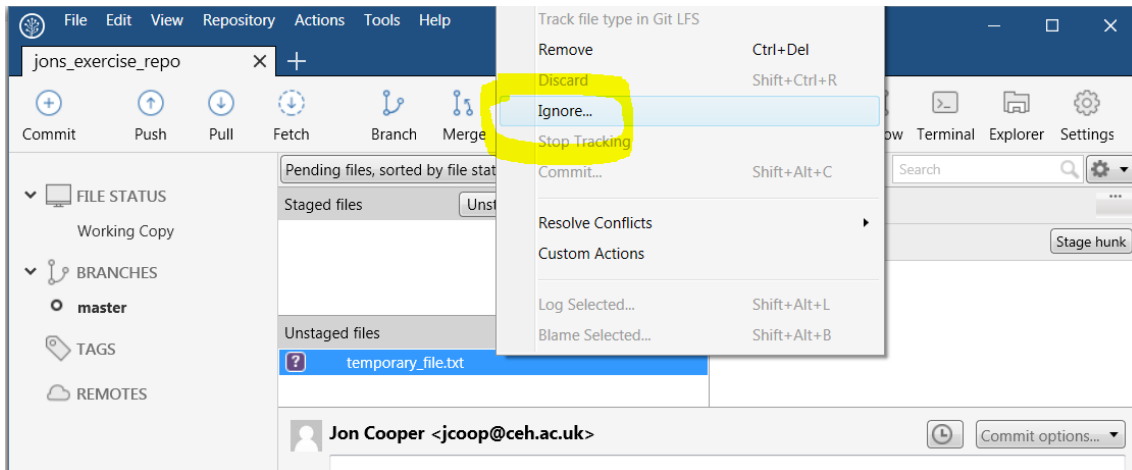


- Cuando necesite sus archivos en el estado al que se refiere la etiqueta, simplemente **verifique** esa etiqueta como lo hizo anteriormente en este ejercicio para una confirmación específica. Sin embargo, por ahora, deja tu repositorio como está.

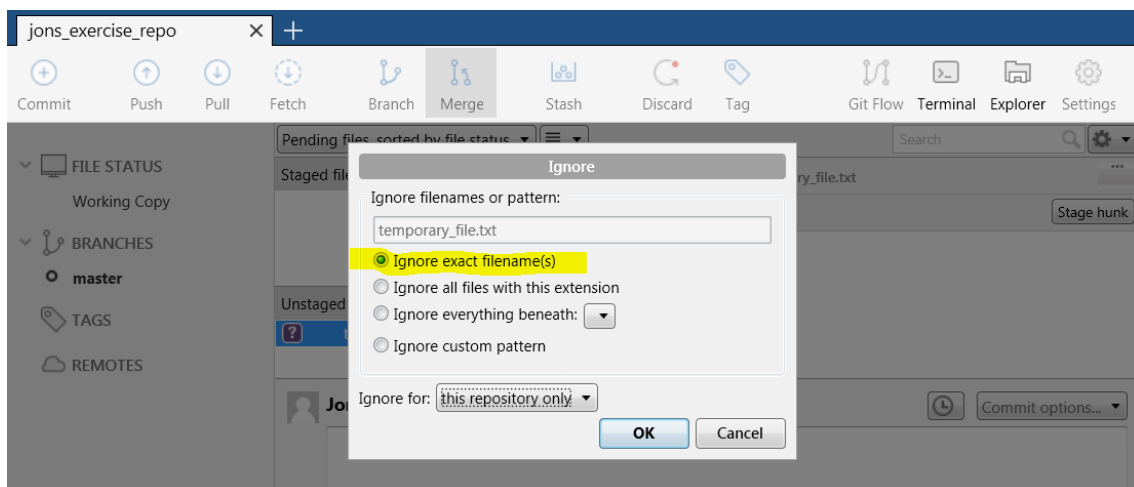
Paso 8. Ignorar archivos

Es muy útil poder excluir algunos archivos del control de versiones, por ejemplo, puede tener otros procesos que generan archivos temporales en su proyecto. Los archivos se ignoran agregando un archivo llamado **.gitignore** a la carpeta raíz de su proyecto. Git ignora los nombres de archivo y los patrones de archivo que se le agregan. SourceTree hace que esto sea fácil de hacer de la siguiente manera.

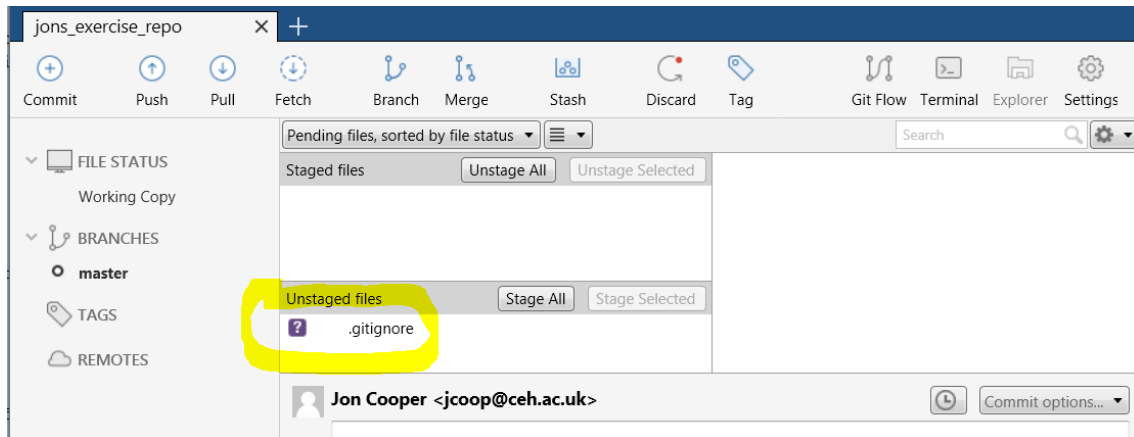
- cree un archivo en la raíz de su proyecto llamado **archivo_temporal.txt** , este es el que va a eliminar del control de versiones.
- En SourceTree, el archivo aparece en el panel **Archivos sin preparar** como sin seguimiento (signo de interrogación morado). Para eliminarlo del control de versiones, haga clic derecho y en el menú contextual seleccione **Ignorar** :



Se le darán varias opciones que le permitirán ignorar varios archivos o solo este. Seleccione **Ignorar nombre(s) de archivo exacto(s)** para ignorar solo este archivo y presione **OK** .



- En el área **Unstaged** el archivo ha desaparecido ya que ahora se está ignorando. Apareció otro archivo, llamado .gitignore, este es solo un archivo que le dice a Git qué ignorar. **Prepare** y **confirme** este archivo con un breve mensaje.



- En el Explorador de Windows, eche un vistazo a la raíz de su proyecto. Verá que el archivo temporal_archivo.txt todavía está allí. Puedes abrirlo y editarlo. Sin embargo, los cambios a este archivo no aparecerán en SourceTree porque ahora le ha dicho a Git que lo ignore.