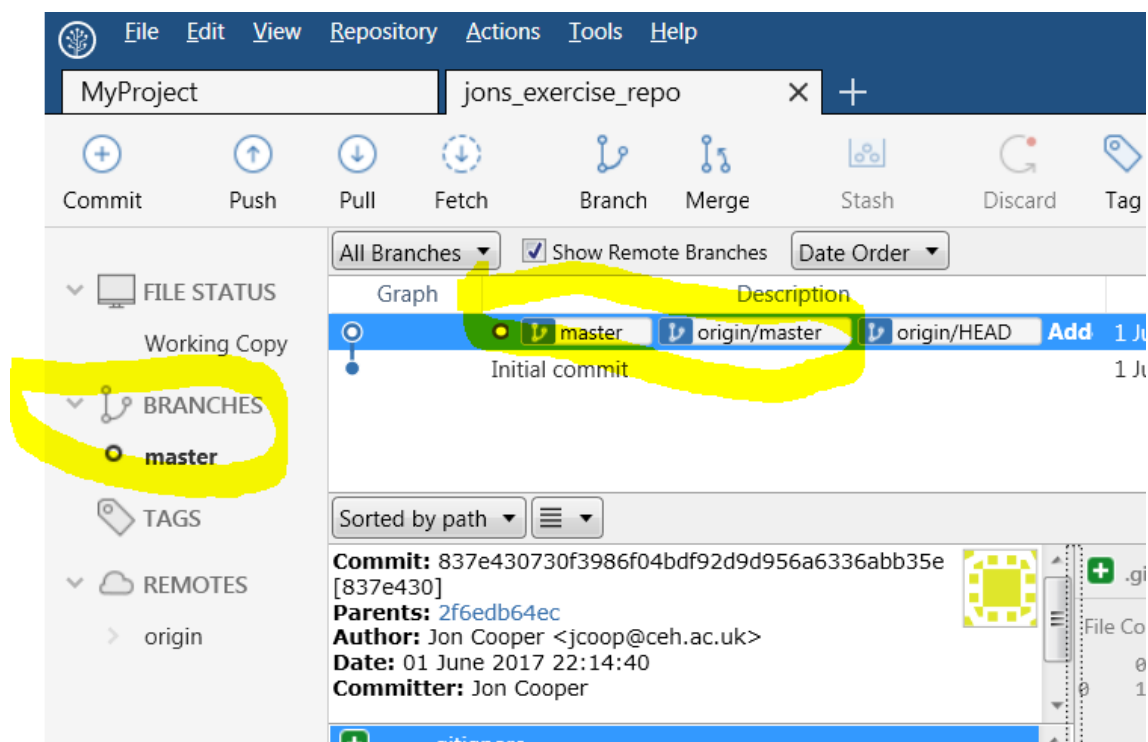


Trabajo en equipo

En los ejercicios anteriores, usted ha sido la única persona que ha trabajado en su repositorio alojado en Github. Esto puede ser suficiente para sus necesidades. Sin embargo, puede ser parte de un equipo en el que todos necesitan editar la misma base de código de forma administrada. En este ejercicio, cada miembro del equipo trabajará en un clon del mismo repositorio de Github y compartirá sus cambios con los demás a través de Github usando acciones **de extracción y inserción**.

Para hacer esto de manera efectiva, también aprenderá sobre **ramificaciones y fusiones**. Hasta ahora, todas sus confirmaciones se han realizado en la rama predeterminada: esta rama se llama **master**. Es posible que ya haya notado **el maestro** en SourceTree como se destaca aquí (en la pestaña **Registro/Historial**):



Sin embargo, es común crear nuevas ramas para trabajos específicos. Esto aísla los cambios de la rama principal y de cualquier otra rama que se haya creado. Solo cuando esté listo, estos cambios se fusionarán desde su nueva rama a la rama maestra (o potencialmente a cualquier rama) a través de una **solicitud de extracción**. Esto es muy útil en proyectos ocupados con múltiples editores y mucho trabajo. Hay más de una forma de crear una rama, en este ejercicio lo harás a través de Github.

En general, aprenderá sobre:

- Creando una sucursal
- Obtener cambios de su equipo
- Resolviendo conflictos
- Crear y aceptar solicitudes de incorporación de cambios para volver a fusionar una rama con la rama principal

Requisitos previos: En línea

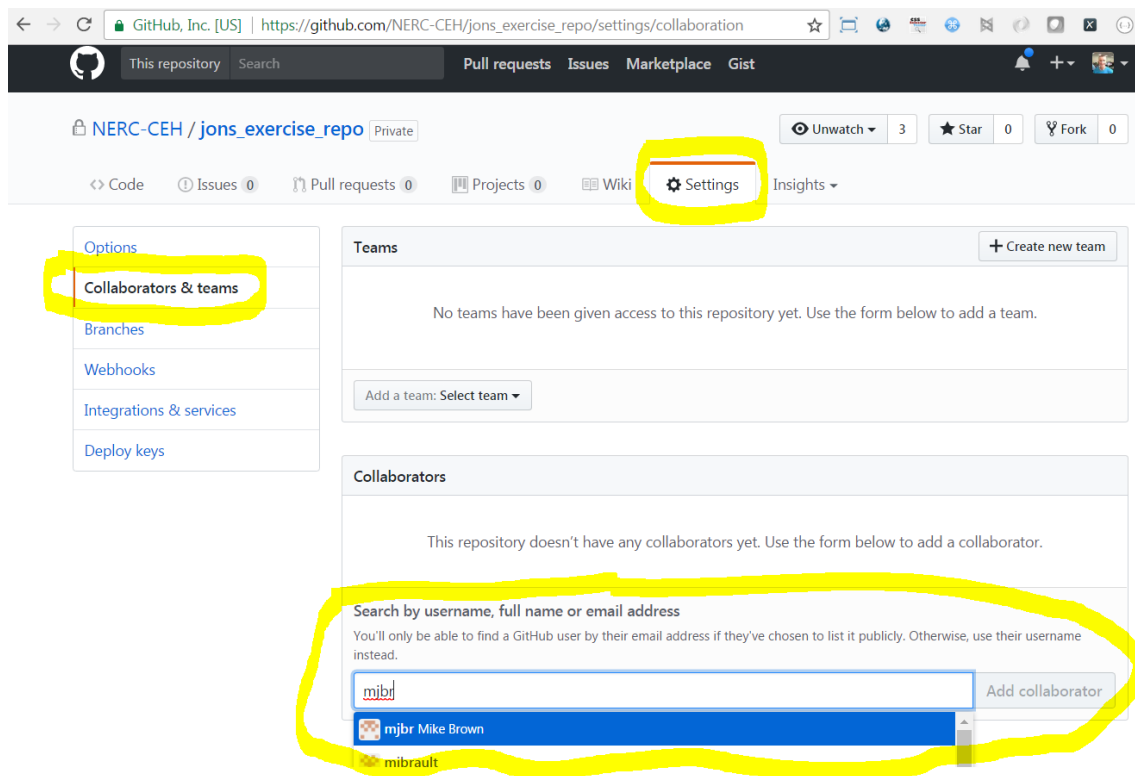
Para participar en este ejercicio necesitas estar en línea. Debería haber al menos wifi disponible durante el taller, así que utilícelo.

Paso 1. Formar un equipo

- Forme equipo con al menos otra persona. Trabajarán juntos para ver cómo sus cambios se afectan entre sí.

Paso 2. Colabora en uno de tus repositorios de Github

- Elija con qué repositorio del Ejercicio 2 van a trabajar todos. El propietario de este repositorio debe agregar a sus compañeros de equipo como **colaboradores** en Github para que todos puedan trabajar en él. El propietario debe ir al repositorio en Github, seleccionar la pestaña **Configuración cerca de la parte superior, luego Colaboradores y equipos** y luego agregar a sus compañeros de equipo usando el cuadro de búsqueda en el panel **Colaboradores** . Recibirán una invitación por correo electrónico que deberán aceptar. Si eso suena complicado, [esta página de ayuda en Github puede ser más clara](#) . Así es como se ve la página para agregar colaboradores:



Paso 3. Repositorio de clones de colaboradores

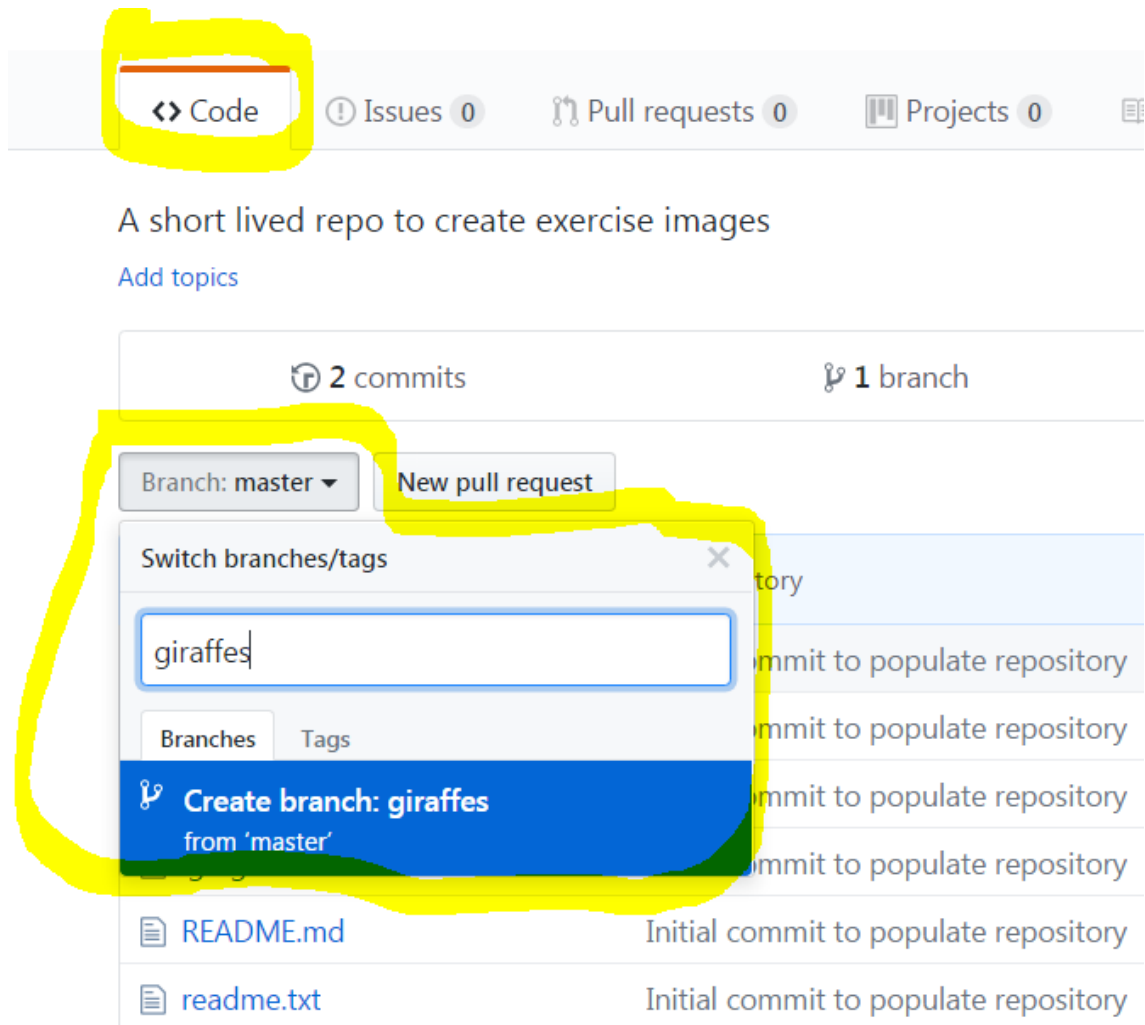
Todo el mundo necesita un clon local del repositorio. El propietario ya debería tener uno. Sin embargo, los colaboradores (es decir, otros miembros del equipo) deben clonarlo desde Github. Siga las instrucciones del [Paso 3 del Ejercicio 2](#) y use la URL del repositorio en el que está colaborando.

- Todos los miembros de su equipo ahora deberían tener un clon local del mismo repositorio listo para realizar ediciones y administrar en SourceTree.

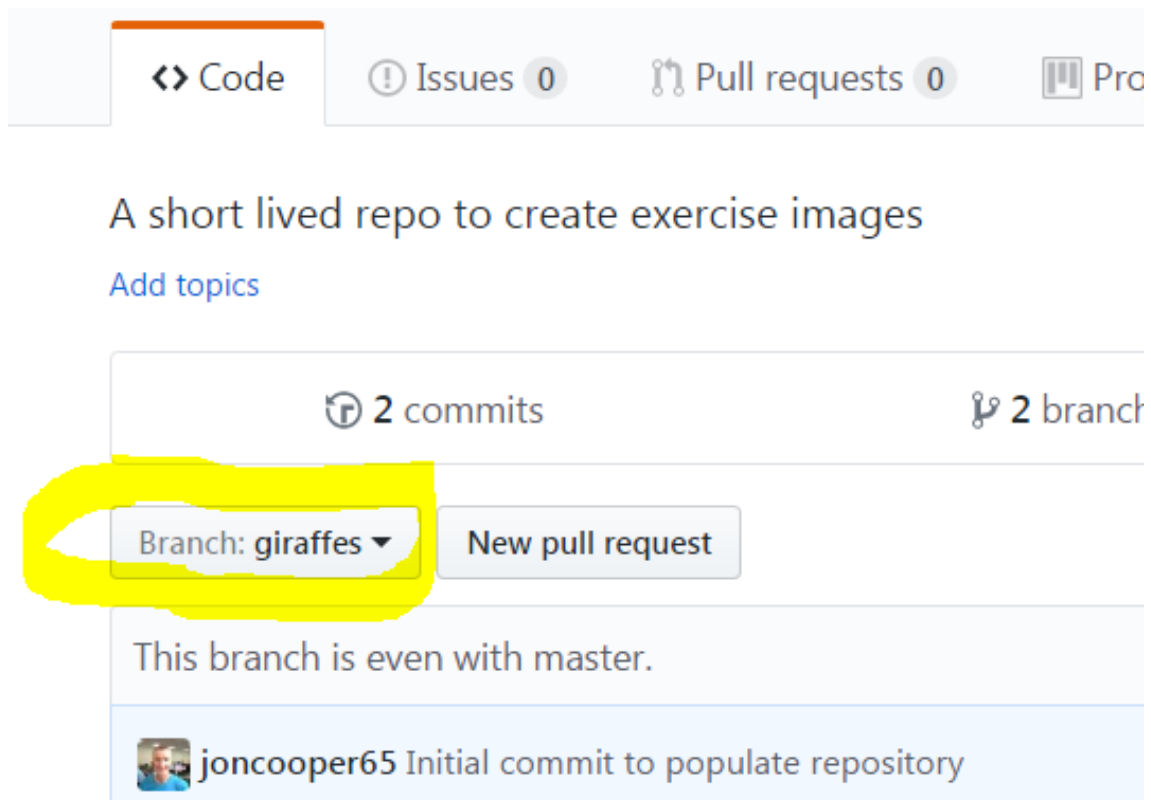
Paso 4. Crea una nueva rama a través de Github

¡El equipo necesita hacer un trabajo urgente para documentar a las jirafas! Ahora una persona debe crear una nueva sucursal de Giraffes en Github, lista para que todo el equipo la **pague** y trabaje en ella.

- una persona del equipo vaya al repositorio en Github, y en la pestaña **Código**, busque el menú desplegable **Sucursal**, **selecciónelo**, **escriba jirafas** en el cuadro y presione Entrar:



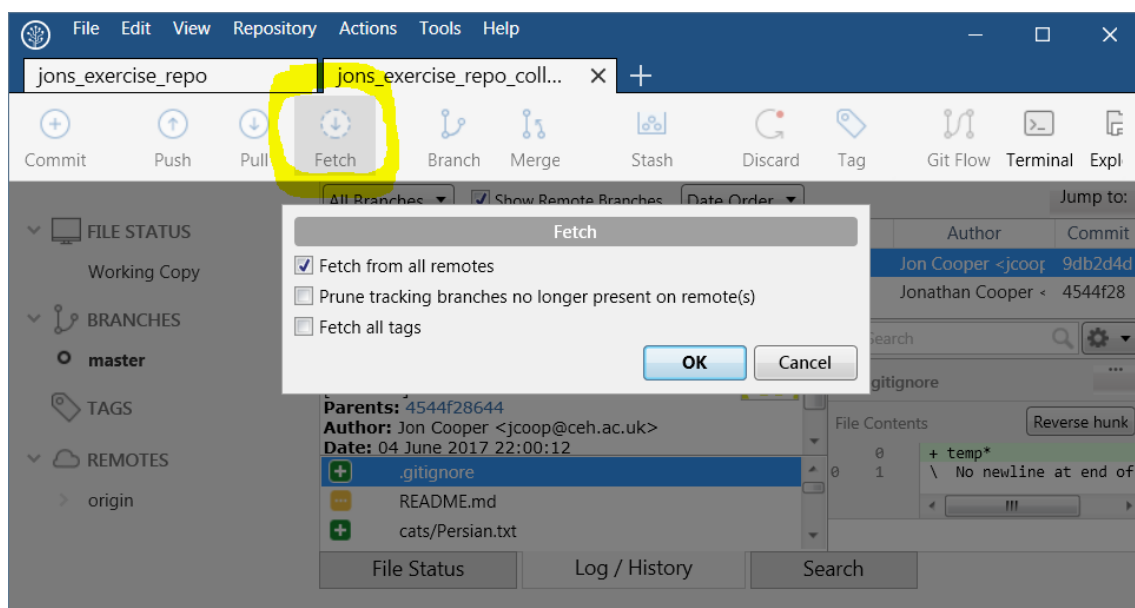
- ahora verás la nueva rama disponible en Github:



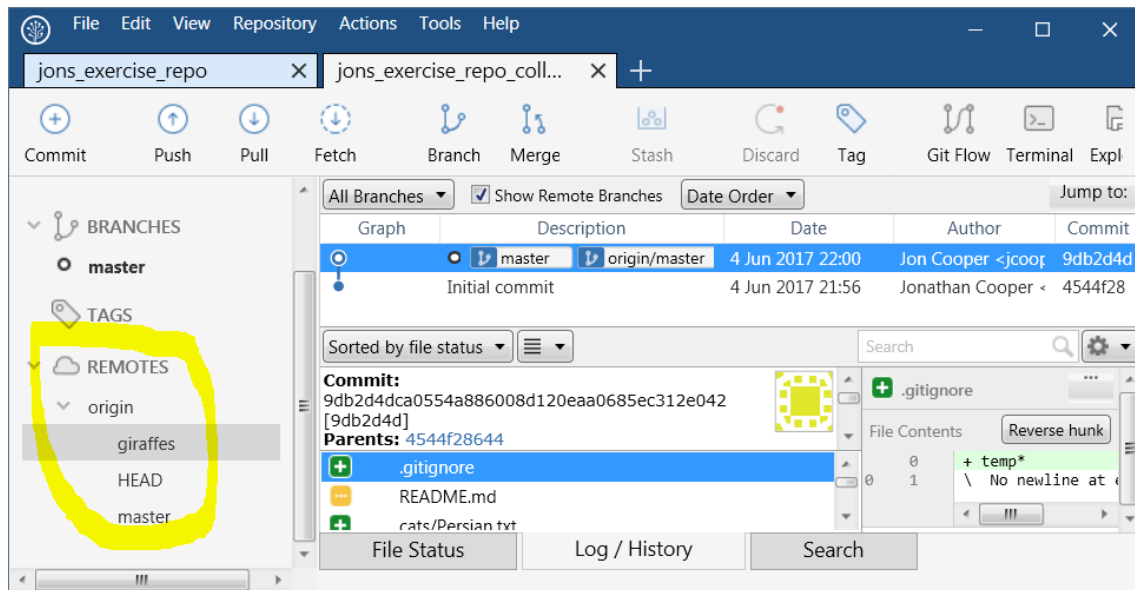
Paso 5. Obtener y verificar la nueva sucursal

Todos ahora necesitan la rama de la jirafa en su repositorio local para poder trabajar en los archivos. Este es un proceso de dos etapas.

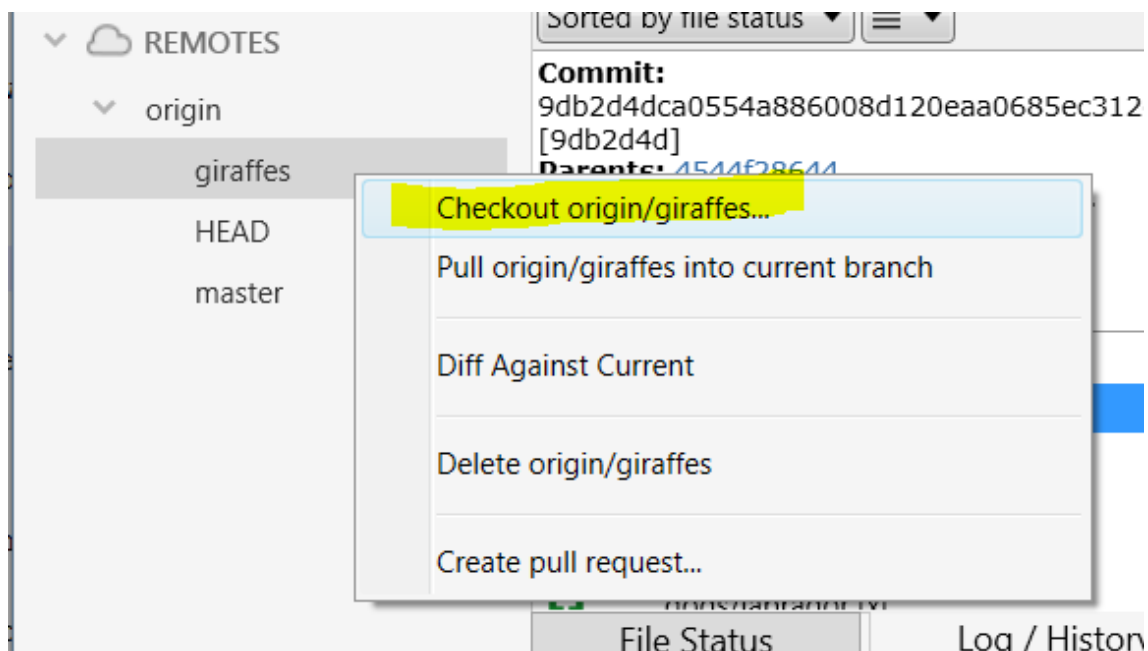
- En primer lugar, con su repositorio abierto en SourceTree, presione el botón **Obtener** y en el diálogo simplemente marque **Obtener desde todos los controles remotos** :



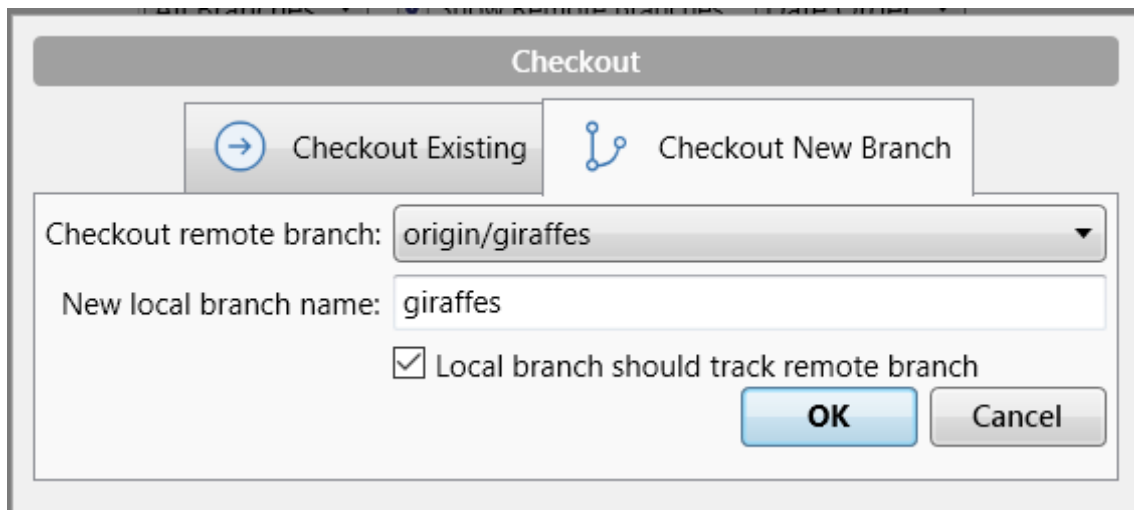
- Esto ha obtenido información sobre su nueva rama, que puede encontrar en el menú desplegable **Remotos** > **origen** a la izquierda, observe que ha aparecido la rama **de jirafas** :



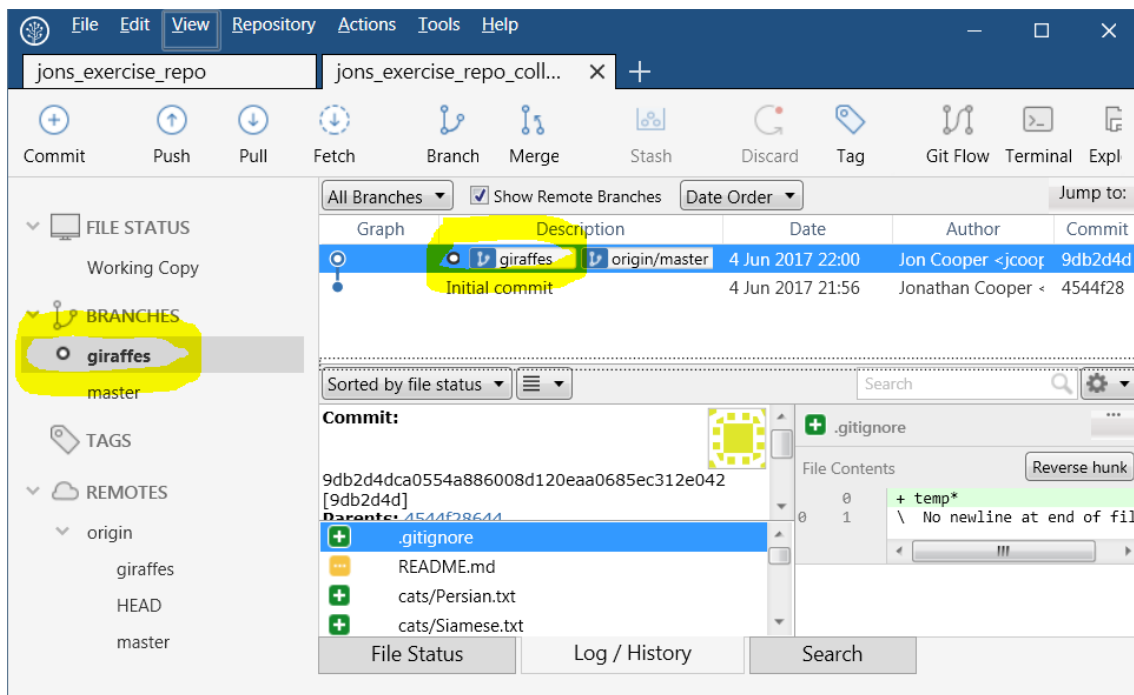
- En segundo lugar, echa un vistazo a la rama de las jirafas haciendo clic derecho sobre ella y seleccionando **Ver origen/jirafas...**



- En el cuadro de diálogo **Pagar** que aparece, seleccione **Pagar nueva sucursal** y mantenga los valores predeterminados como se muestra a continuación:



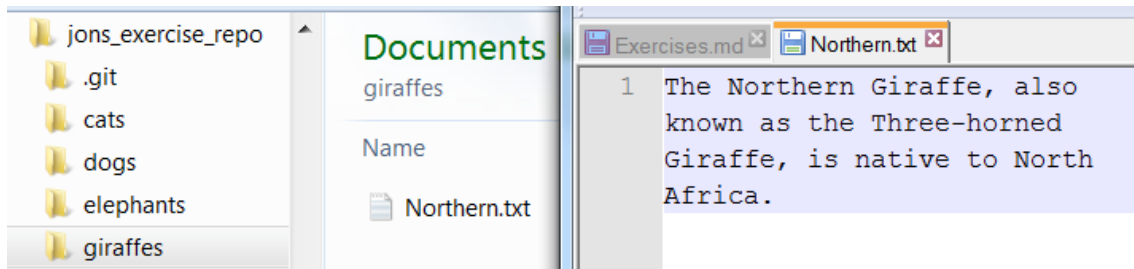
- Ahora todos tendrán la rama de la jirafa en su repositorio local lista para usar, así:



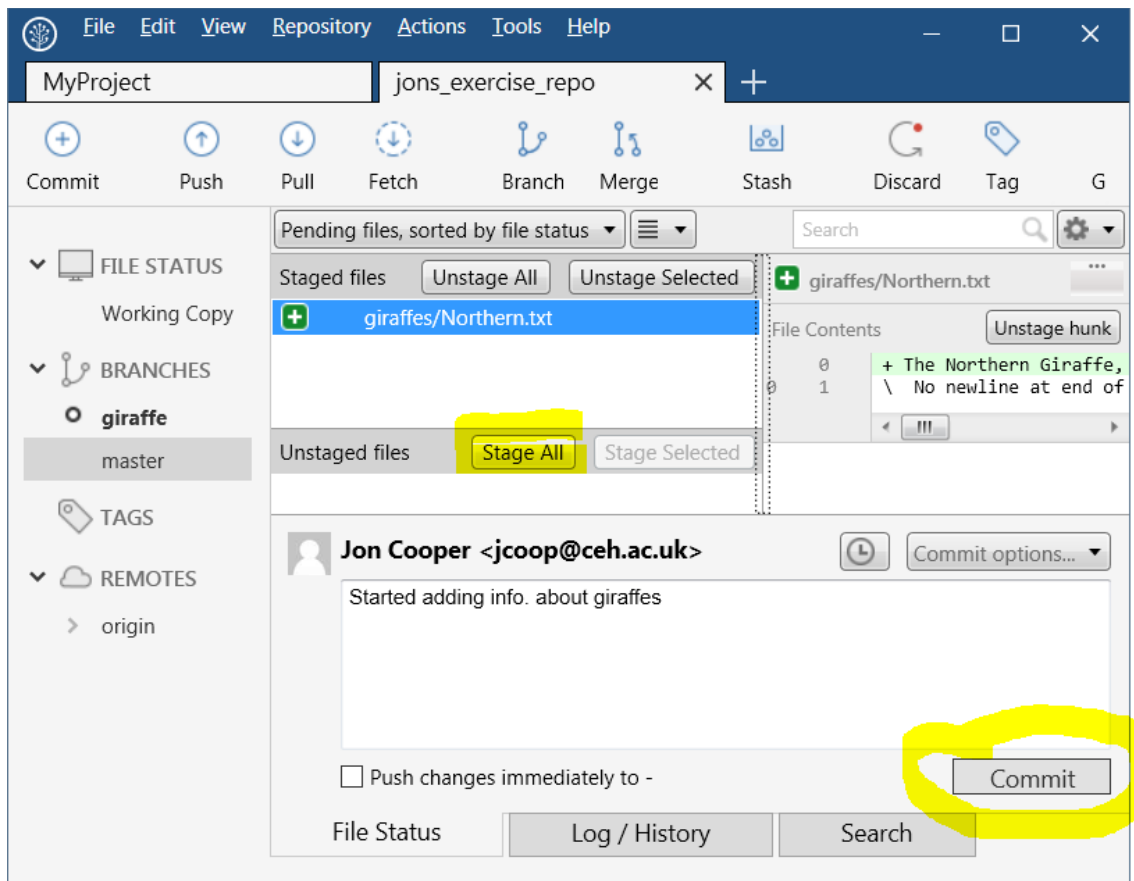
Paso 6. Empujar y tirar de los cambios

Es hora de empezar a trabajar en equipo. Ahora agregará algo de contenido a la rama de jirafas que todos han revisado. Dos personas trabajarán en partes separadas y las subirán a Github.

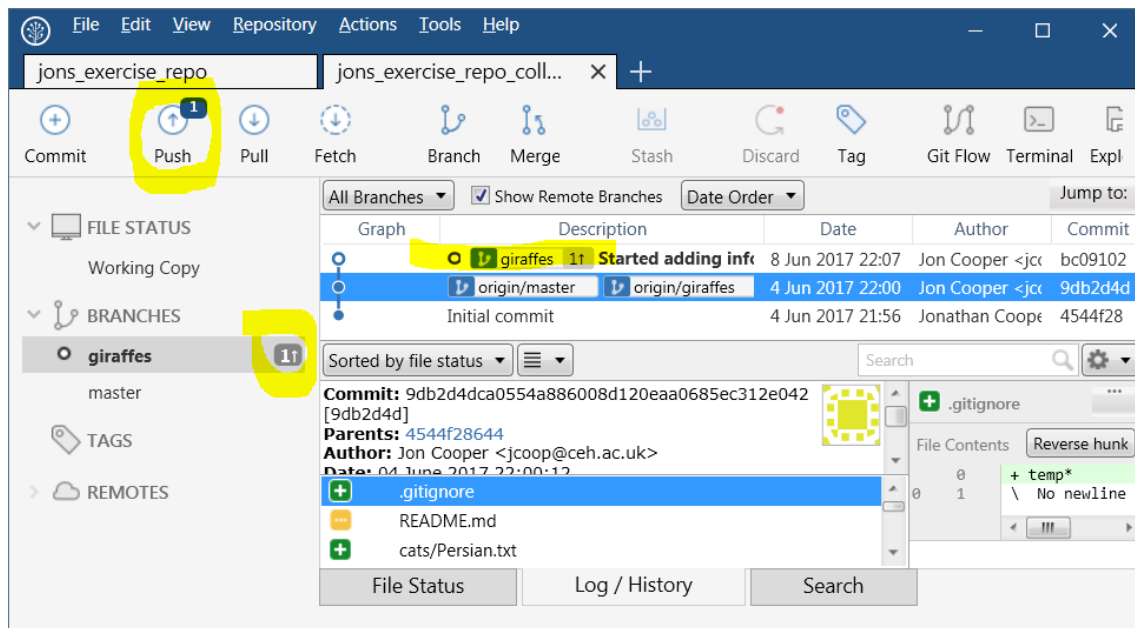
- **Persona 1:** en el Explorador de Windows, cree una carpeta en la raíz de su proyecto llamada **jirafas**, dentro de esa carpeta cree un nuevo archivo de texto llamado Northern.txt que contenga algo de texto sobre Northern Giraffe.



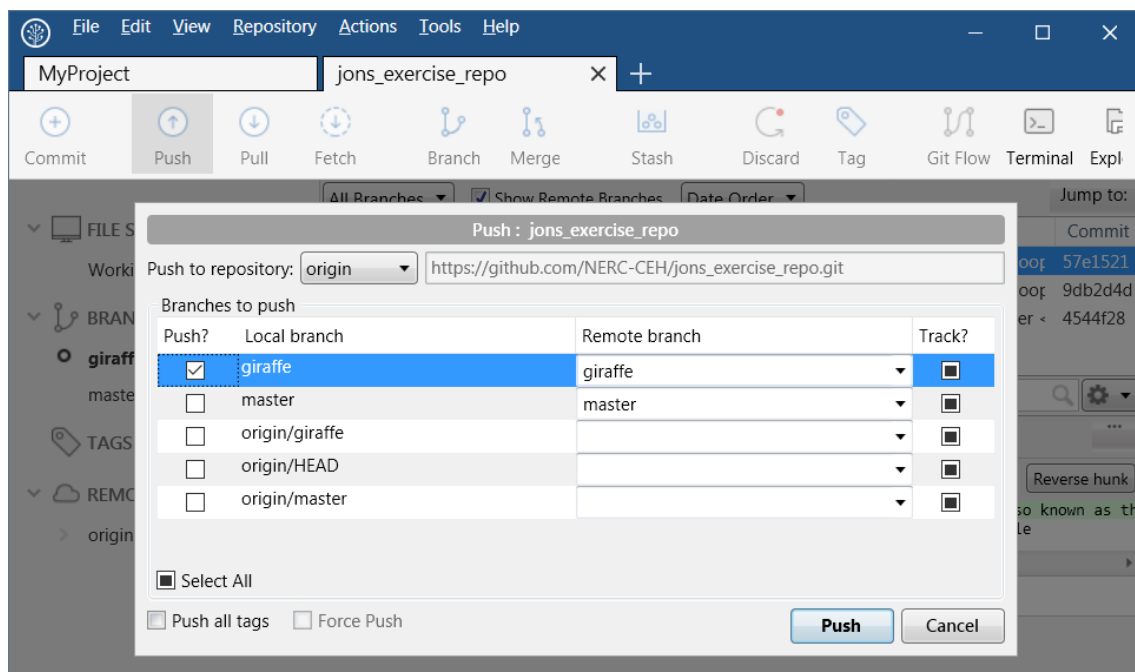
- **Persona 1:** en SourceTree, organice y confirme sus ediciones con un breve mensaje:



- **Persona 1:** En su vista **de Registro/Historial**, verá la rama Giraffe delante de la rama Master, ya que tiene cambios que Master no tiene. También verá que ha aparecido un número 1 en varios lugares que indica que hay una nueva confirmación que no se ha subido a Github:




- **Persona 1:** ahora **envíe** sus cambios a Github como lo ha hecho antes, presionando el botón **Push** . Esta vez se le preguntará qué ramas empujar, ya que solo ha trabajado en Giraffe, luego marque esa como se muestra a continuación.



- Todos los miembros del equipo pueden revisar los cambios en Github en el navegador si lo desean. En la pestaña **Código** , verá su nueva carpeta de jirafa entre las demás.

Your recently pushed branches:

 **giraffes** (less than a minute ago)


Branch: giraffes ▾








New pull request

Create new file

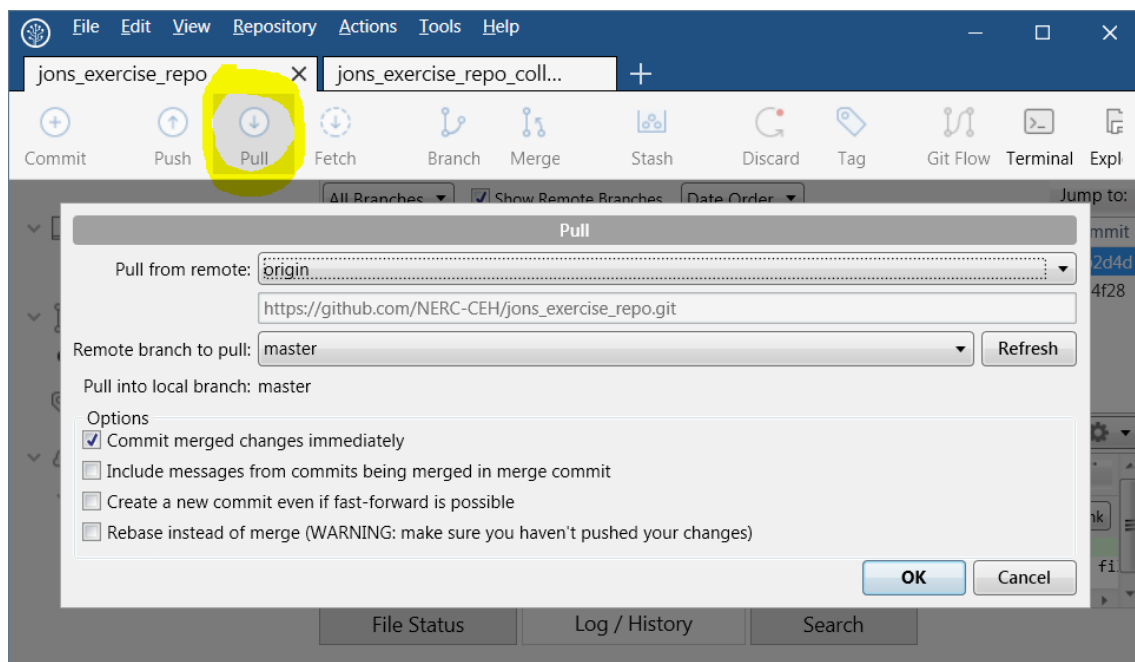
Upload file

This branch is 1 commit ahead of master.

 joncooper65 Started adding info. about giraffes

 cats	Initial commit to populate repository
 dogs	Initial commit to populate repository
 elephants	Initial commit to populate repository
 giraffes	Started adding info. about giraffes
 .gitignore	Initial commit to populate repository
 README.md	Initial commit to populate repository
 readme.txt	Initial commit to populate repository

- **Persona 2:** es tu turno de hacer un trabajo. En primer lugar, **extraiga** el trabajo de jirafa de sus colegas, use los valores predeterminados en el cuadro de diálogo Extraer que aparece:



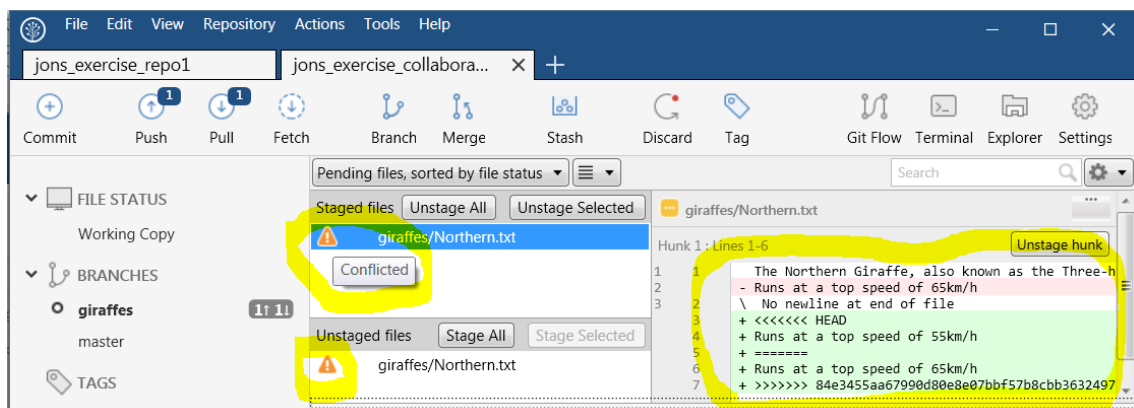
- **Persona 2:** Ahora debería ver la nueva carpeta de jirafas que contiene Northern.txt, si no la ve, asegúrese de haber revisado la rama de jirafas y no estar en la rama principal. Simplemente agregue una nueva línea de texto a giraffes/Northern.txt, por ejemplo, 'Pueden correr a 55 km/h'. Prepare y confirme su edición, y luego suba a Github.
- Ahora debería tener la idea de que usted y su(s) colega(s) pueden **presionar** para compartir cambios y **tirar** para recibirlos y, por lo

tanto, todos se mantienen sincronizados entre sí. De esta manera, todos contribuyen a la rama del repositorio hasta que se realiza el trabajo. Pero, ¿qué sucede si dos personas editan de forma independiente el mismo texto? Eso es el próximo paso.

Paso 7. Resolución de conflictos

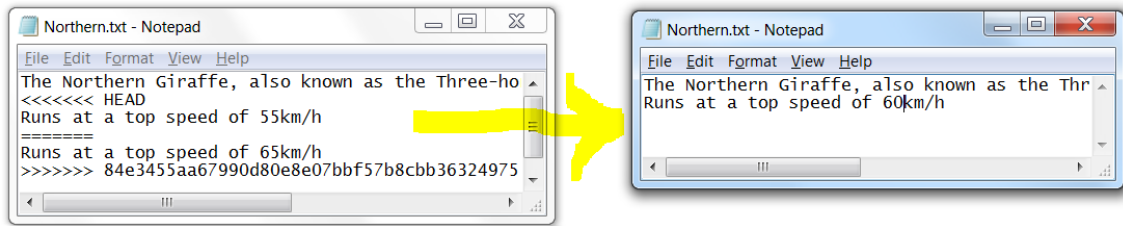
En este paso aprenderá a resolver los conflictos que surgen cuando dos personas editan de forma independiente la misma parte de un archivo con texto diferente. Crearán el conflicto al editar el mismo texto, una vez que uno de ustedes envíe su cambio a Github, el otro verá errores de conflicto cuando intenten enviar su cambio. Usted decidirá cuál debería ser el texto correcto y enviará la resolución a Github. Así que empieza por crear el conflicto...

- **Dos miembros del equipo** deben acordar editar la misma línea del mismo archivo (por ejemplo, giraffes/Northern.txt), pero agregar un texto ligeramente diferente.
- **Persona 1:** prepara, confirma y envía tus cambios a Github.
- **Persona 2:** después de que la persona 1 haya enviado su cambio, haga un **stage** y **confirme**, pero esta vez haga un **pull** antes de **empujar** para bajar su edición. Obtendrá los siguientes triángulos de advertencia en SourceTree que dicen **En conflicto** cuando pase el mouse sobre ellos. Si selecciona su archivo de problema, el texto conflictivo se muestra en el panel de la derecha en verde con texto adicional insertado que intenta mostrarle lo que está sucediendo.

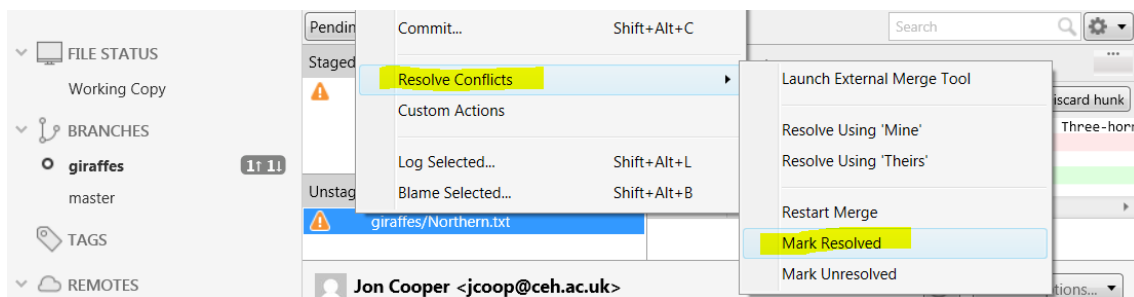


- Entonces, ¿qué está pasando y cómo lo solucionas? Como puede ver en el ejemplo, se ha agregado un texto diferente sobre la velocidad a la que corren las jirafas. El texto en conflicto ahora está bordeado por cheurones («< >>»). ¡Discutan y acuerden cuál debería ser el texto correcto! Luego, **la Persona 2** abre el archivo que contiene el conflicto en un editor y simplemente reemplaza todo lo que se muestra como un conflicto con el

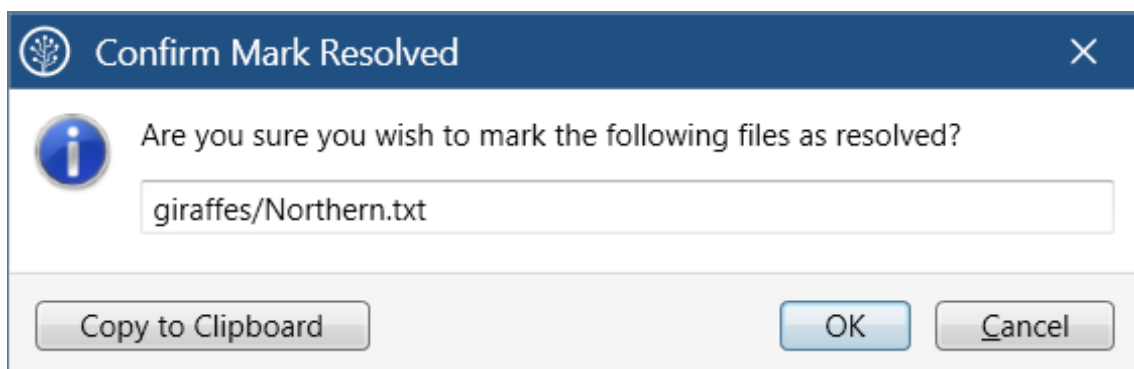
texto correcto, incluidos los cheurones, los signos de igual y el gran número largo, así es como se ve antes y después:



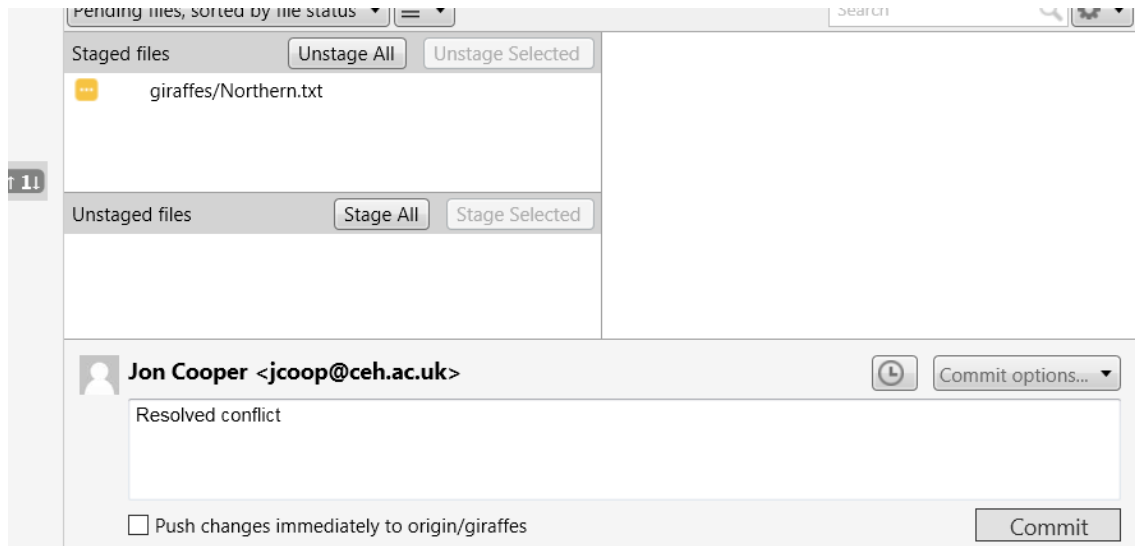
- De vuelta en SourceTree, aún tendrá los triángulos de advertencia de conflicto al lado del archivo. Debe informar a SourceTree que ha resuelto el conflicto, lo que hace haciendo clic con el botón derecho en el nombre del archivo y, en el menú contextual, seleccione **Resolver conflictos > Marcar resuelto**.



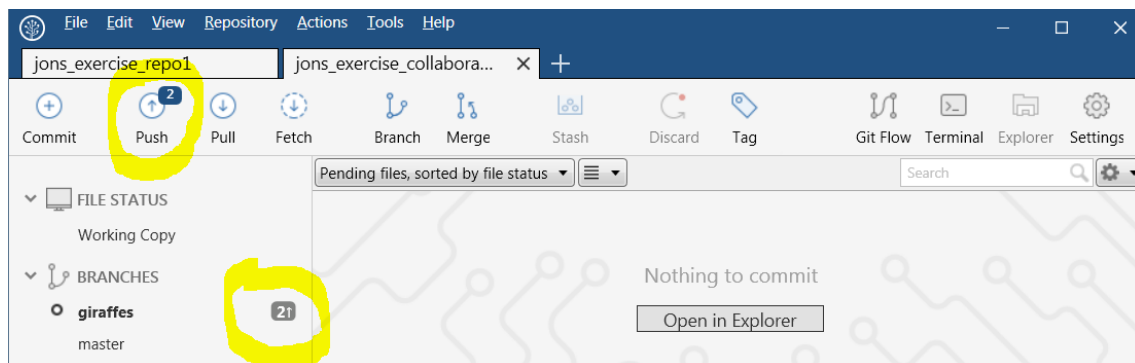
- Acepte el panel de información **Confirmar Marcar Resuelto** que aparece:



- Confirma tus cambios resueltos



- Finalmente, debe tener dos compromisos para enviar, presione el botón Push para enviarlos a Github:

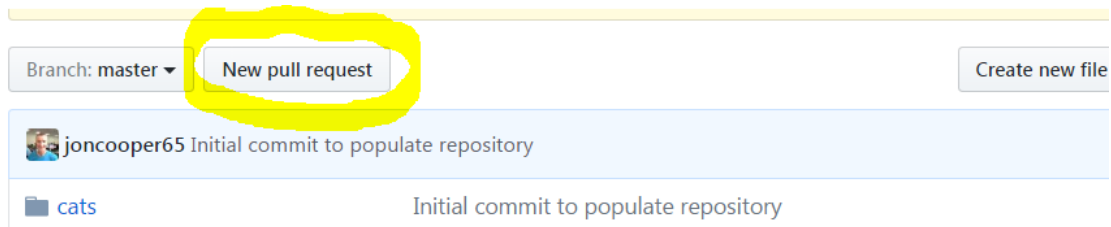


- Agregar una herramienta de resolución de conflictos externa a la que SourceTree puede transferir puede hacer que ese proceso sea mucho más fácil, pero este ejercicio no cubre eso (todavía). El proceso que acabas de hacer te permite editar manualmente cualquier conflicto, por lo que tienes los conocimientos suficientes para salir del apuro.

Paso 8. Solicitud de extracción

El equipo ha terminado todo el trabajo de la sucursal de jirafas. Es hora de volver a fusionarlo con la rama **Maestra**. Vamos a hacer esto mediante la emisión de una **solicitud de extracción** en Github. Una solicitud de extracción involucra a otros miembros del equipo para revisar el trabajo antes de que la rama se fusione con la rama maestra. Puede involucrar hilos de discusión y cambios adicionales. Cuando están satisfechos con nuestros cambios, la rama se fusiona con la maestra. Es un paso de control de calidad que ayuda a garantizar que la rama de destino (generalmente la maestra) solo contenga trabajo aprobado.

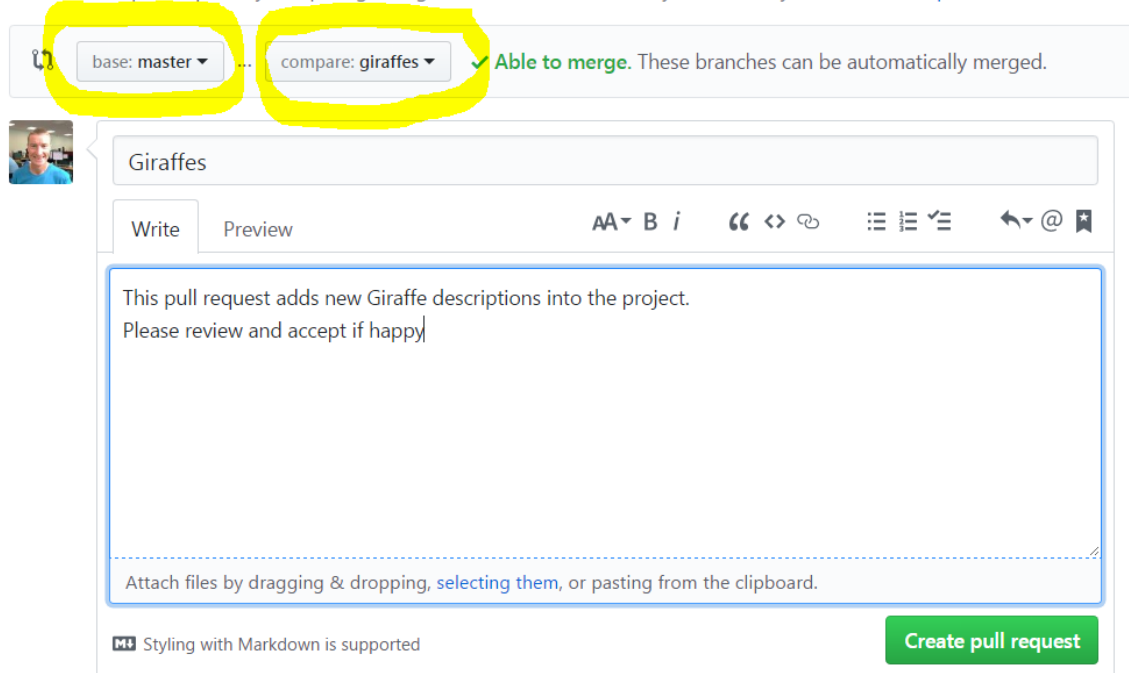
- Solo un miembro del equipo debe crear la solicitud de extracción de la siguiente manera
- en Github, en la página principal del repositorio del equipo, en la pestaña **Código** , presione el botón **Nueva solicitud de extracción** :



- En los dos desplegados que aparecen, selecciona **base:** **maestro** y **compara:** **jirafas** . La rama de destino para la solicitud de extracción es la rama **base** . Debería ver un mensaje verde **Capaz de combinar** que indica que la rama de jirafas se fusionará con la maestra sin conflictos. Agregue una descripción de lo que se trató el trabajo y presione **Crear solicitud de extracción**

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



- Todos los colaboradores del repositorio reciben un correo electrónico para informarles que se ha enviado una solicitud de extracción. También verán la solicitud de fusión pendiente cuando vayan a la página principal del repositorio en Github y seleccionen la pestaña **Solicitudes de extracción** . Uno de ustedes debe echar un vistazo, revisar el trabajo y

cuando esté satisfecho con él, presione **Merge pull request** , seguido de **Confirm merge** :

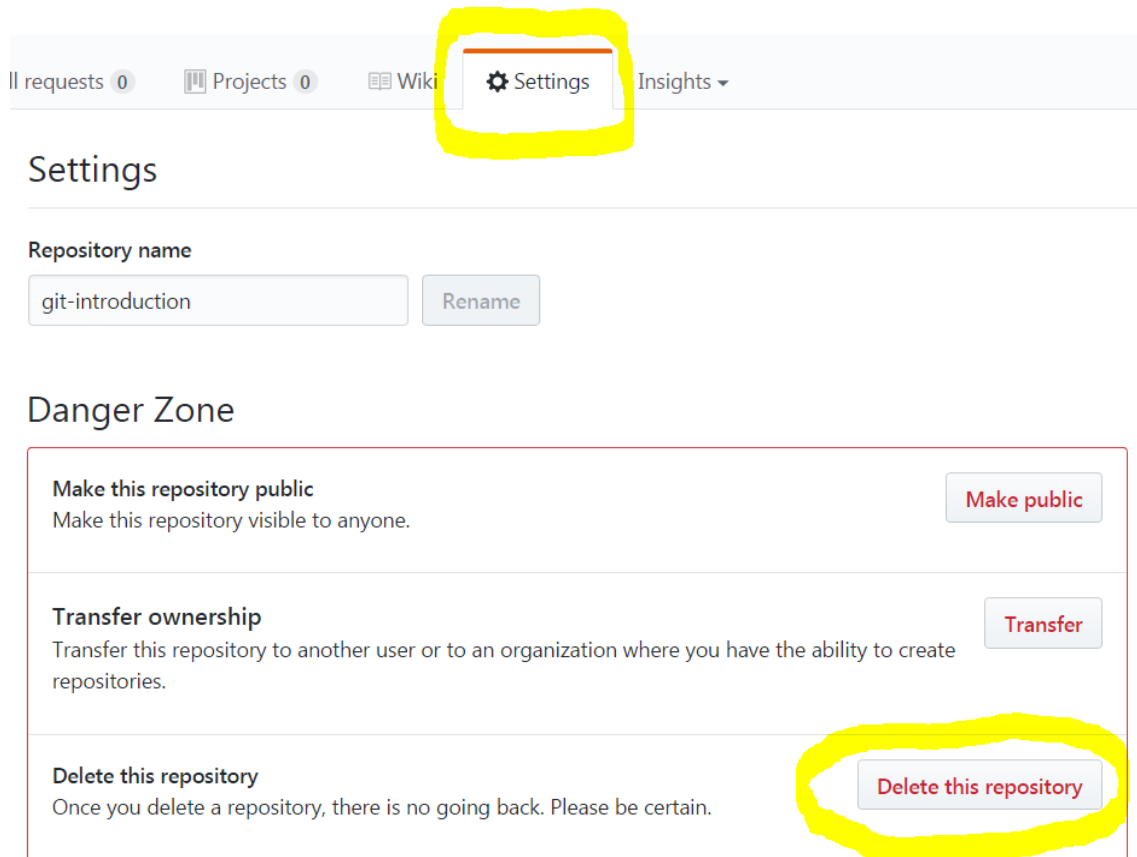
The screenshot shows the GitHub pull request interface for a repository named "Giraffes #1". At the top, there is a navigation bar with tabs for "Code", "Issues 0", "Pull requests 1", "Projects 0", "Wiki", "Settings", and "Insights". Below the navigation bar, the title "Giraffes #1" is displayed. A green button labeled "Open" is next to the text "joncooper65 wants to merge 4 commits into master from giraffes". Below this, there are tabs for "Conversation 0", "Commits 4", and "Files changed 1". The main content area shows a comment from joncooper65, posted 2 minutes ago, stating: "This pull request adds new Giraffe descriptions into the project. Please review and accept if happy". Below the comment, there is a list of commits added by joncooper65: "Added giraffe info", "Added giraffe speed", "Added fastest speed", and "Resolved conflict". At the bottom, there is a message: "Add more commits by pushing to the giraffes branch on NERC-CEH/jons_exercise_repo.". A green box with a checkmark icon contains the text: "This branch has no conflicts with the base branch. Merging can be performed automatically." Below this, there is a green button labeled "Merge pull request" and a link: "You can also open this in GitHub Desktop or view command line instructions."

- Informará que la solicitud de incorporación de cambios se ha fusionado correctamente y el creador de la solicitud de fusión será notificado por correo electrónico.

Paso 9. Ejercicios terminados, limpia Github si has terminado con ellos

- Esto concluye los ejercicios de control de versiones. Normalmente mantendrá su repositorio funcionando durante la duración de su proyecto e incluso más. Sin embargo, es muy posible eliminar los repositorios que posee. En unos días después del taller, eliminaremos los repositorios de prueba que ha creado hoy en la organización NERC-CEH. Esto le da tiempo para revisarlos fuera del taller si lo desea. Sin

embargo, no dude en eliminar su repositorio ahora para ayudarnos. Encontrará el botón **Eliminar este repositorio** en la **Zona de peligro** en la pestaña **Configuración** de la página principal de su repositorio, así:



Esperamos que estos ejercicios le hayan resultado útiles y que siga aprendiendo más sobre el control de versiones con Git a medida que lo incorpora a sus procesos de trabajo.