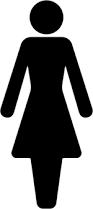
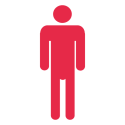
PRÁCTICA DE GIT CON NETBEANS

La siguiente práctica está relacionada con el control de versiones utilizando Git.

Dos desarrolladores, Daniel, Azahara y DanielC están colaborando para desarrollar un programa en Java.

Daniel.  Azahara.  DanielDC

**Commit**: Guardar los cambios en tu repositorio Local.

**Push:** subir los cambios al repositorio Remoto.

**Pull:** Sirve para bajar las últimas versiones subidas al repositorio Remoto.

**Fetch:** Permite bajar una rama temporal para actualizar.

**Merge:** Fusionar las ramas con la principal.

**Branch**: Rama para trabajar sin machacar un proyecto.

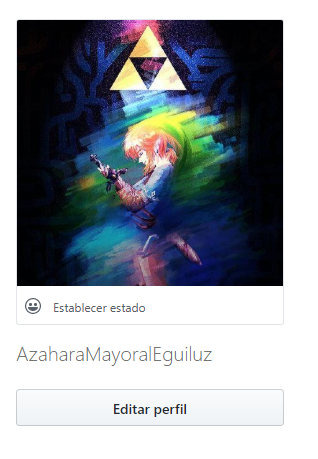
**Tag:** etiqueta.

**Master:** Rama principal.

Para ello, utilizan Git para el control de versiones.



Daniel Azahara DanielDC



Ambos utilizan la versión integrada en Netbeans.

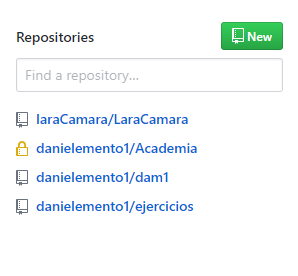


Para llevar a cabo la programación en equipo.

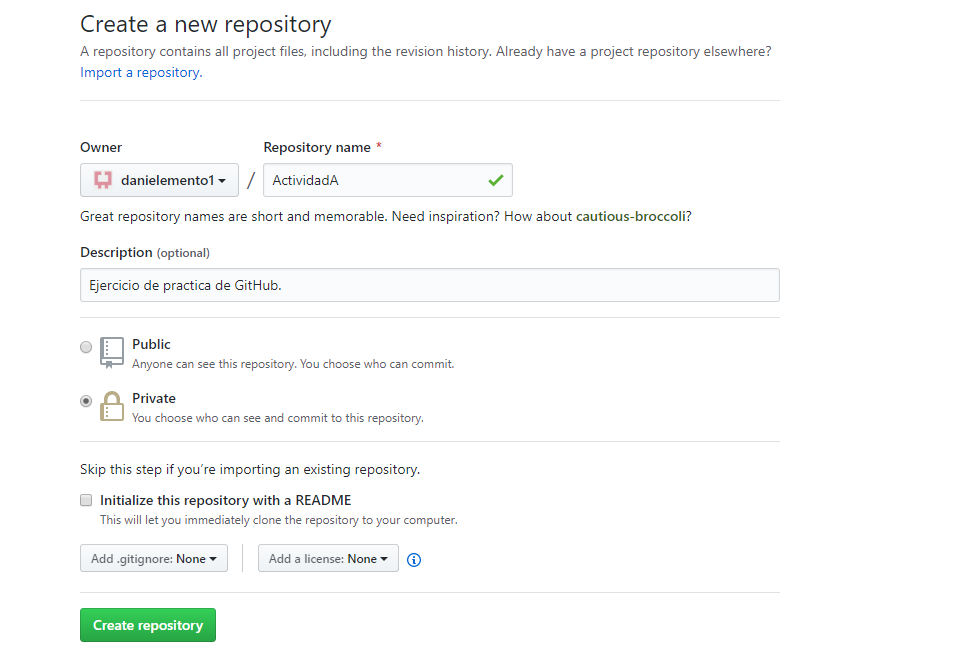


Han creado un repositorio remoto en GitHub llamado ACTIVIDAD A continuación se relatarán una serie de pasos que han ido siguiendo ambos durante el desarrollo.



Daniel entra en su cuenta de GitHub, y crea el repositorio, pulsando new. 

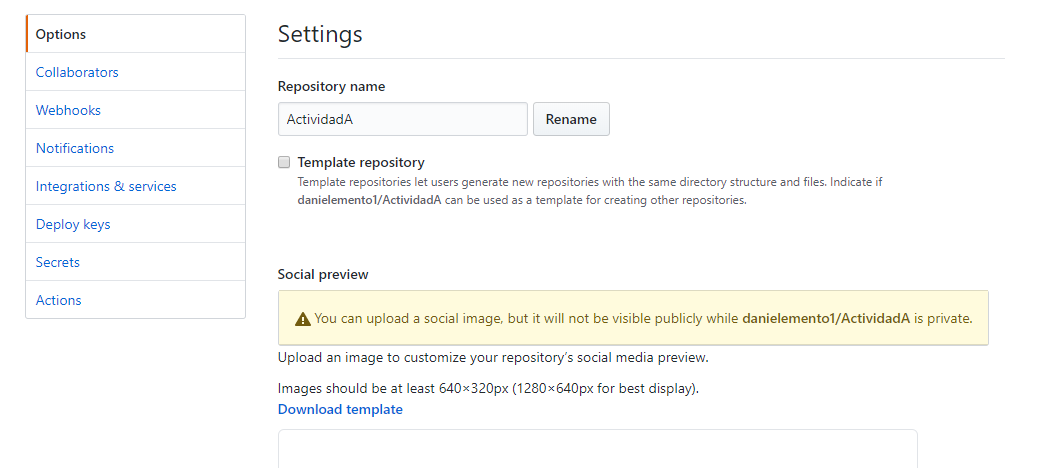
Le da nombre al repositorio, y elige el nivel de acceso: Privado para que solo pueda acceder el creador y los colaboradores



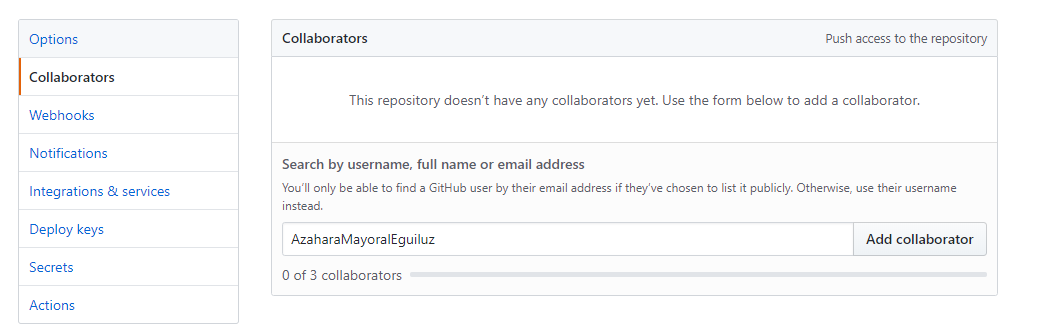
Salta la siguiente pantalla:



Daniel clica en la pestaña Settings:

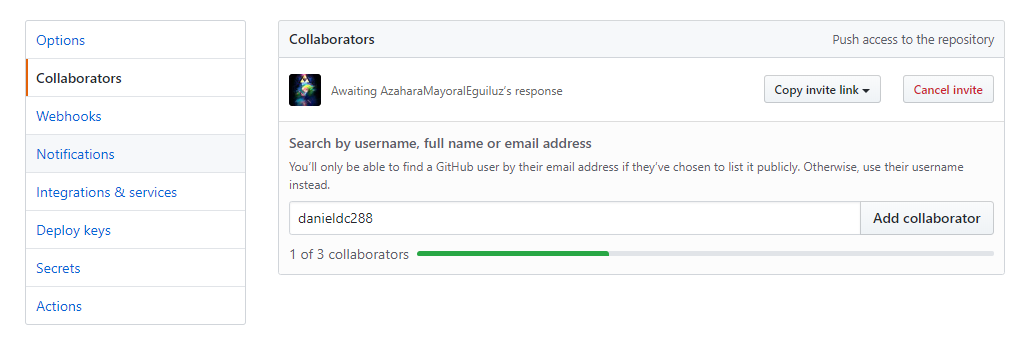


Y dentro de options, pincha en pestaña Collaborators:

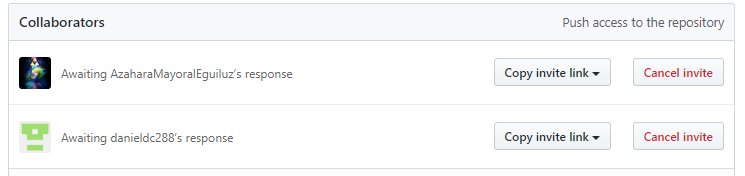


Añade un nuevo colaborador introduciendo su nombre de usuario de GitHub o su correo electrónico.

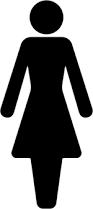
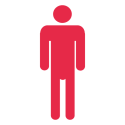
Realiza la misma operación para añadir otro colaborador:



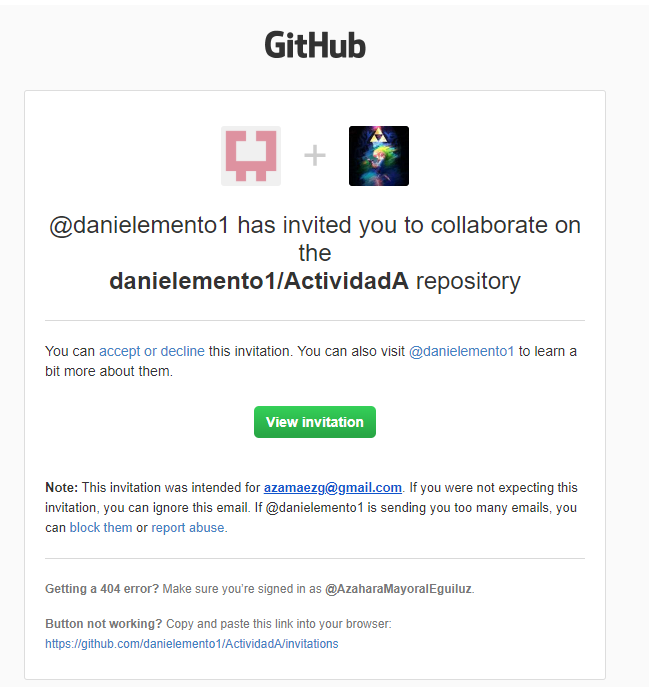
En esta pantalla se muestra los colaboradores a los que les hemos enviado una invitación para colaborar en el proyecto y si han aceptado o no han aceptado aun la propuesta.



Cada colaborador debe aceptar la invitación enviada por el creador del repositorio, este mismo puede cancelar la invitación.

Azahara.  DanielDC

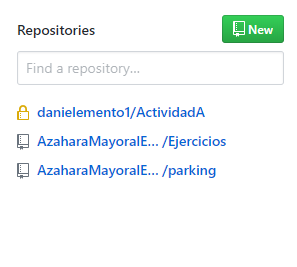
Azahara y DanielDC han recibido un correo con la invitación de colaboración, que deben aceptar para poder participar. 



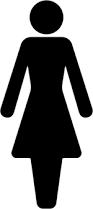
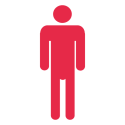
Se da a ver la invitación, en este punto de acepta.

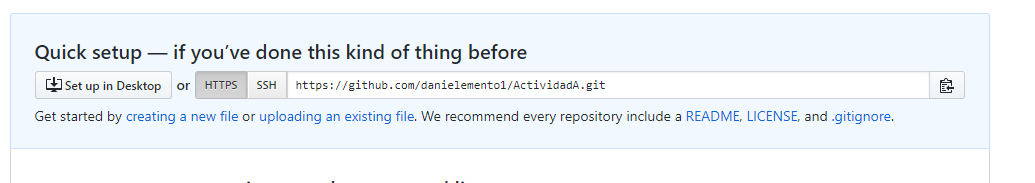


Una vez aceptada se encuentra en tus propios repositorios:



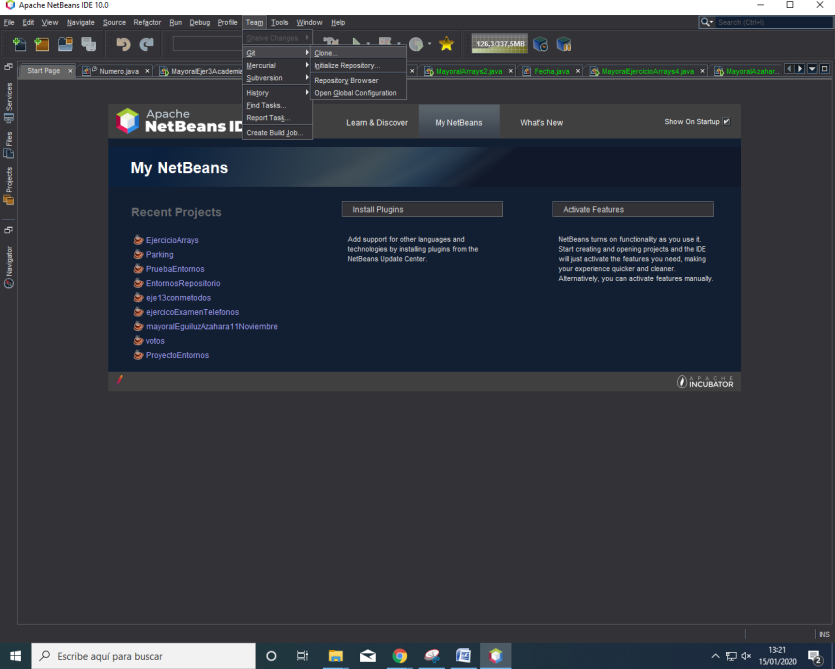
• Lo primero que hacen Daniel, Azahara y DanielDC van a clonar el repositorio remoto cada uno en su equipo.

Daniel.  Azahara.  DanielDC



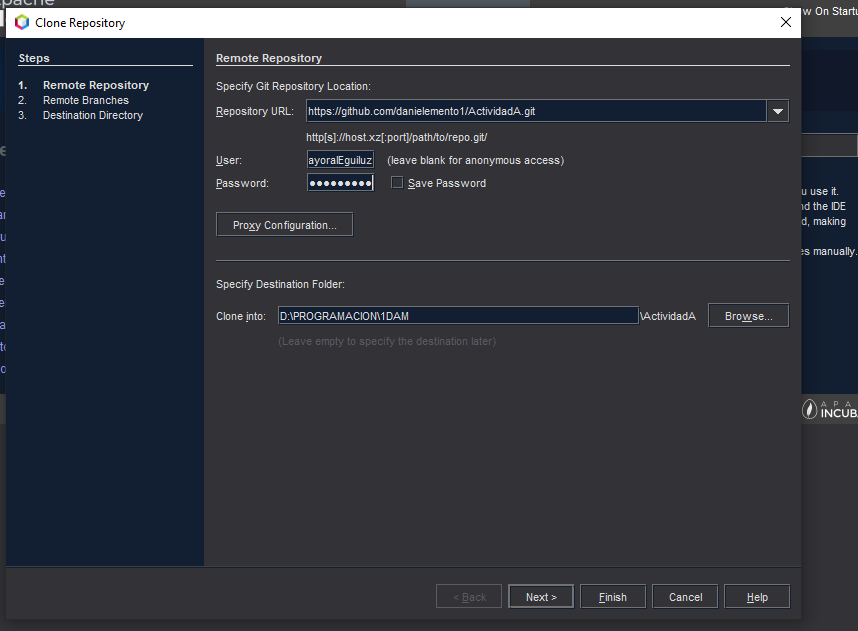
Cada uno de ellos tiene que copar la dirección:

Para clonar el repositorio se ha de entrar en Team, GIT y Clone:

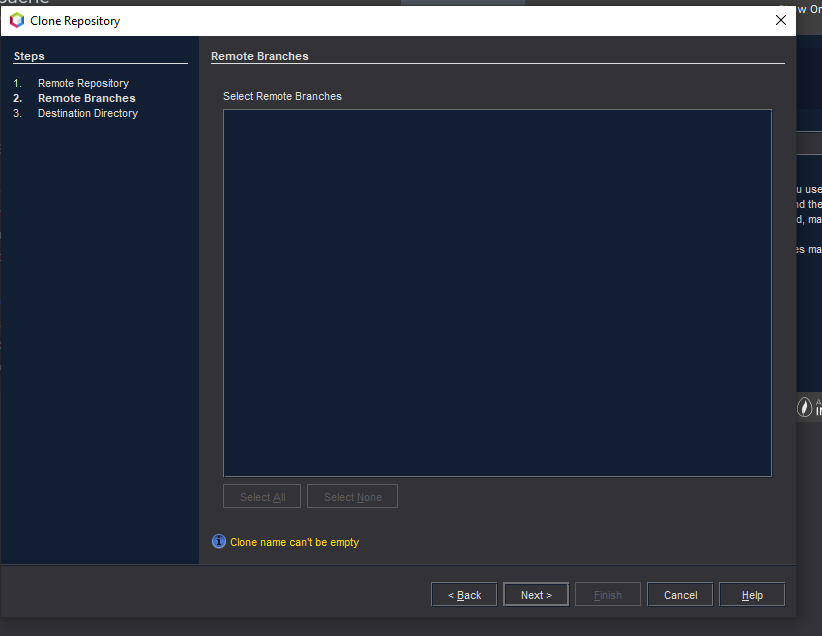


Sale la siguiente pantalla en la cual hay que copiar la dirección.

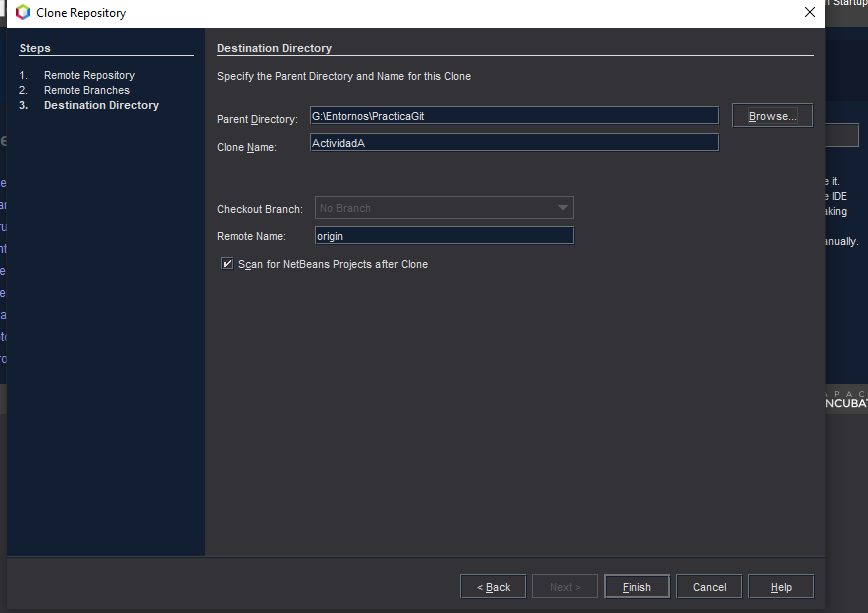
Meter el usuario y la contraseña del mismo.



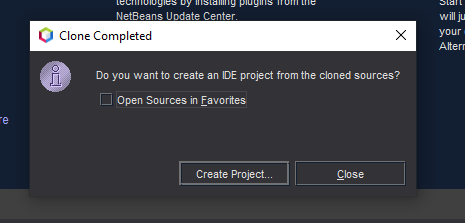
Una vez aquí se copian las ramas (en este caso no hemos creado nada con anterioridad)



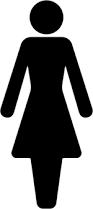
Destino del directorio para repositorio local:



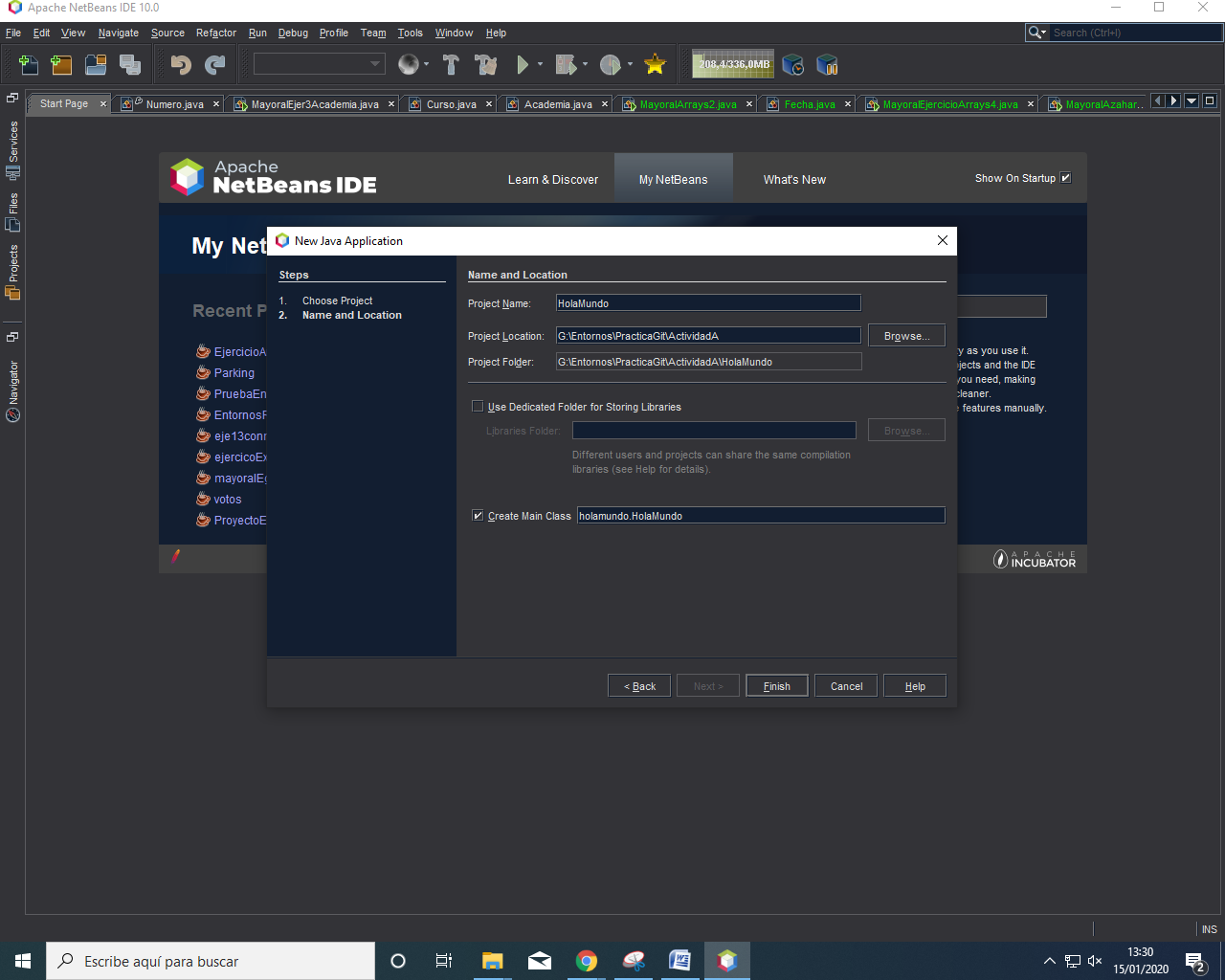
Salta esta pantalla, para crear el proyecto.



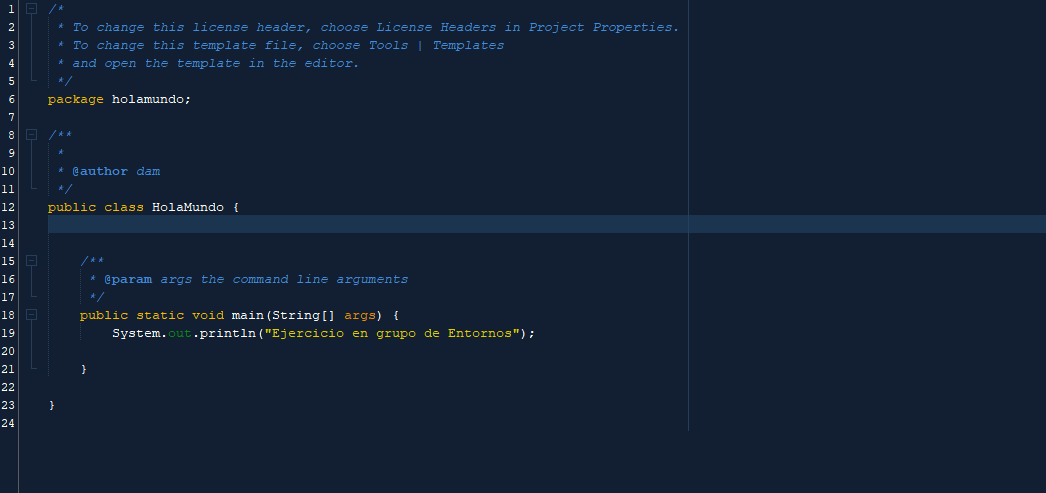
• Una vez terminado el proceso, Azahara crea un proyecto llamado HolaMundo en el repositorio local.

 Azahara.

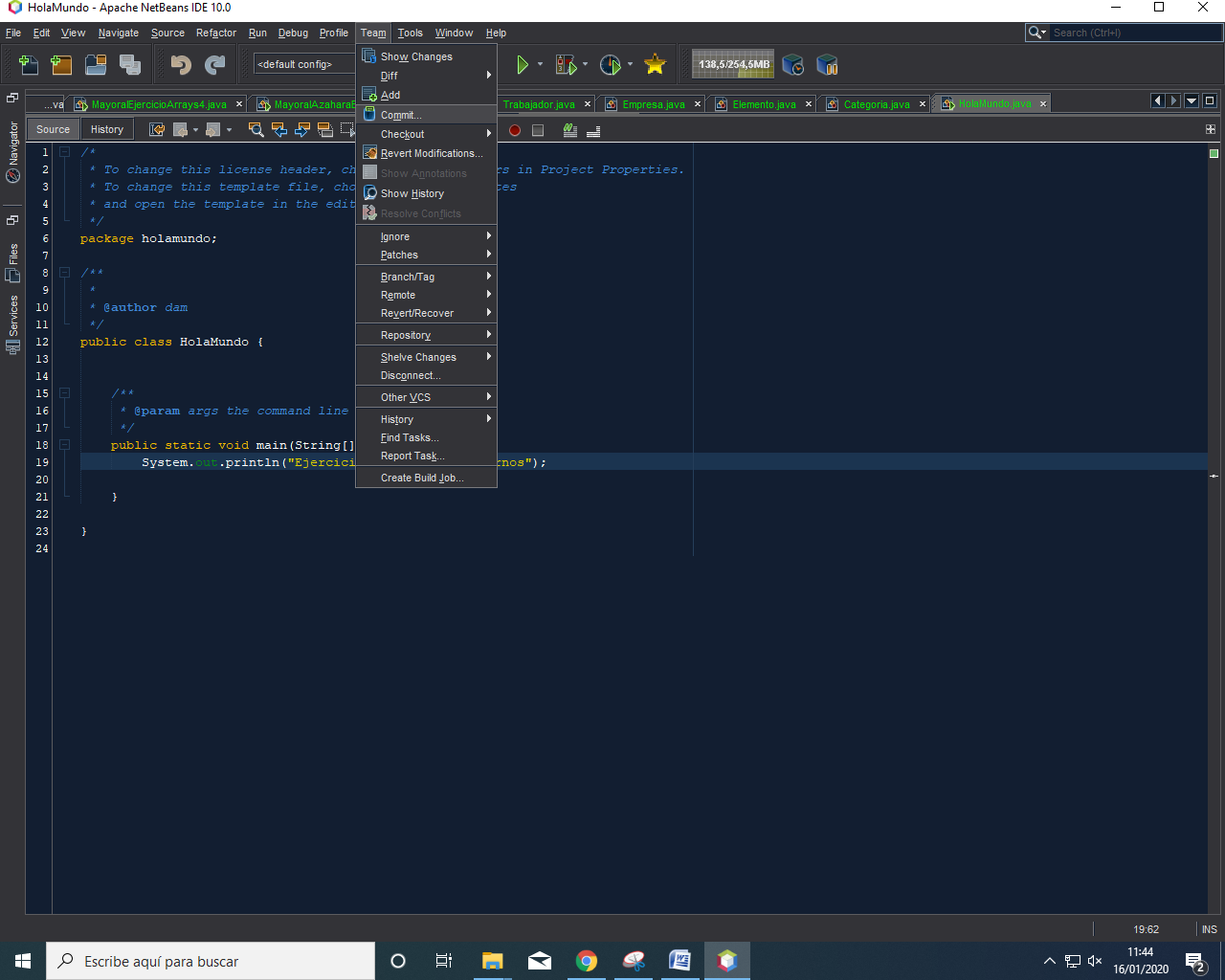
Azahara crea el proyecto HolaMundo:



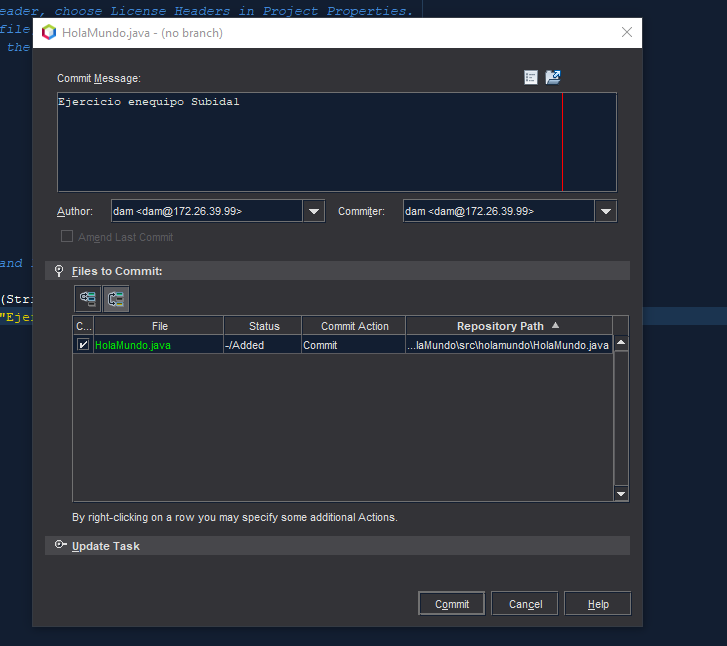
• Se crea un archivo HolaMundo.java, el cual editará y subirá al repositorio remoto.



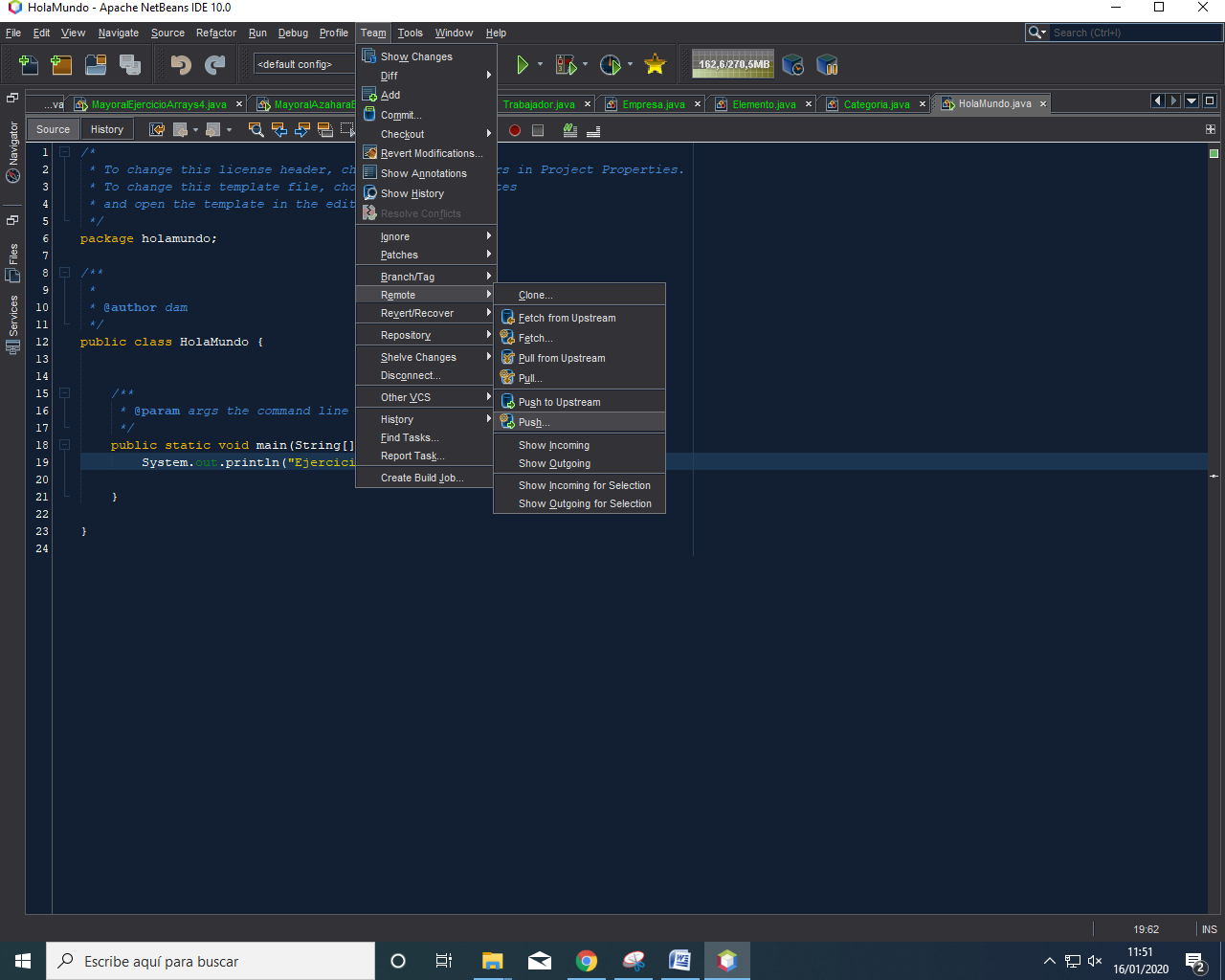
Azahara ha añadido la línea “Ejercicio en grupo de Entornos” y para que sus compañeros la vean en GitHub, tiene que realizar un Commit.



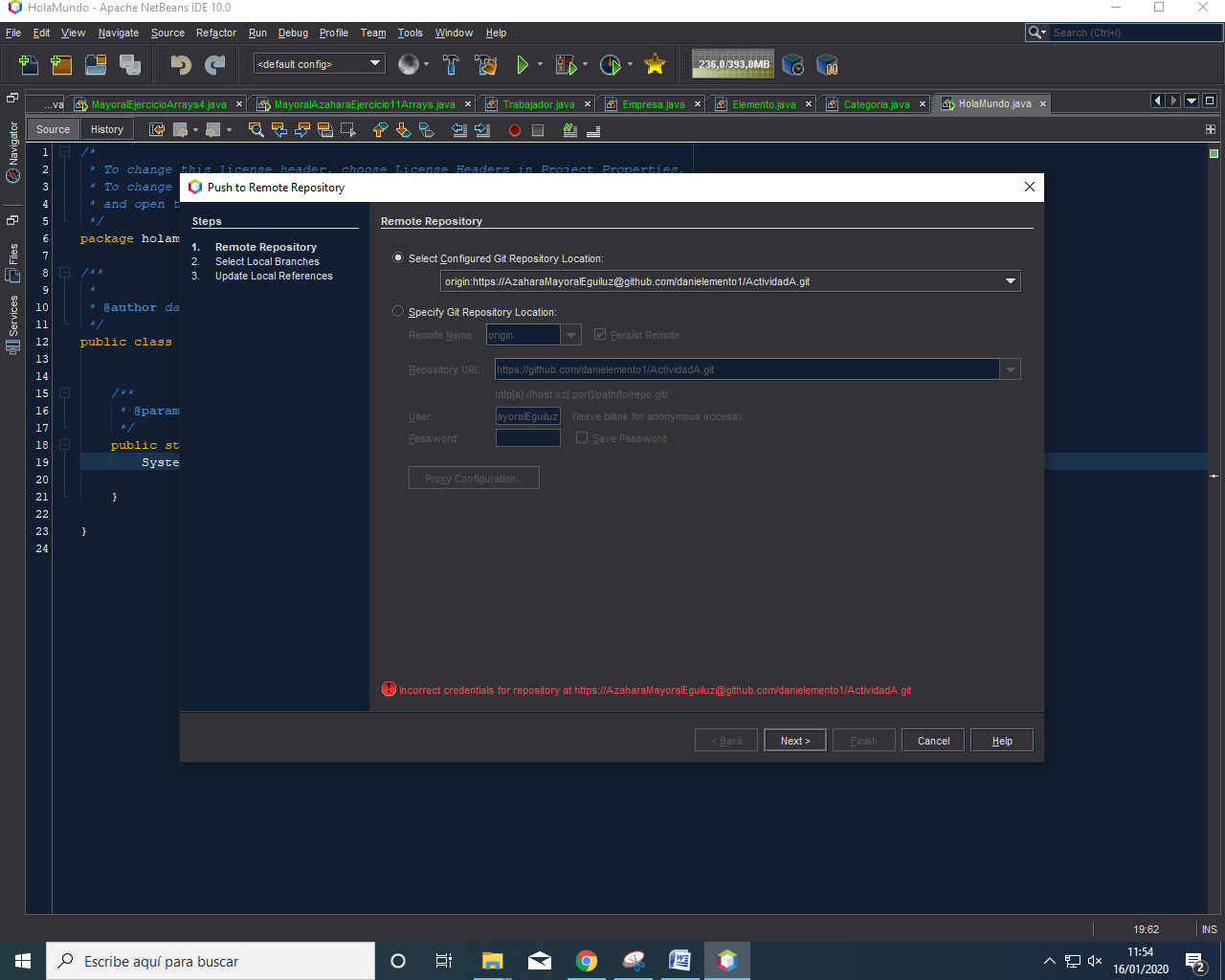
Añade el nombre con el que quieres designar al cambio.



Para subirlo a GitHub tiene que realizar un Push

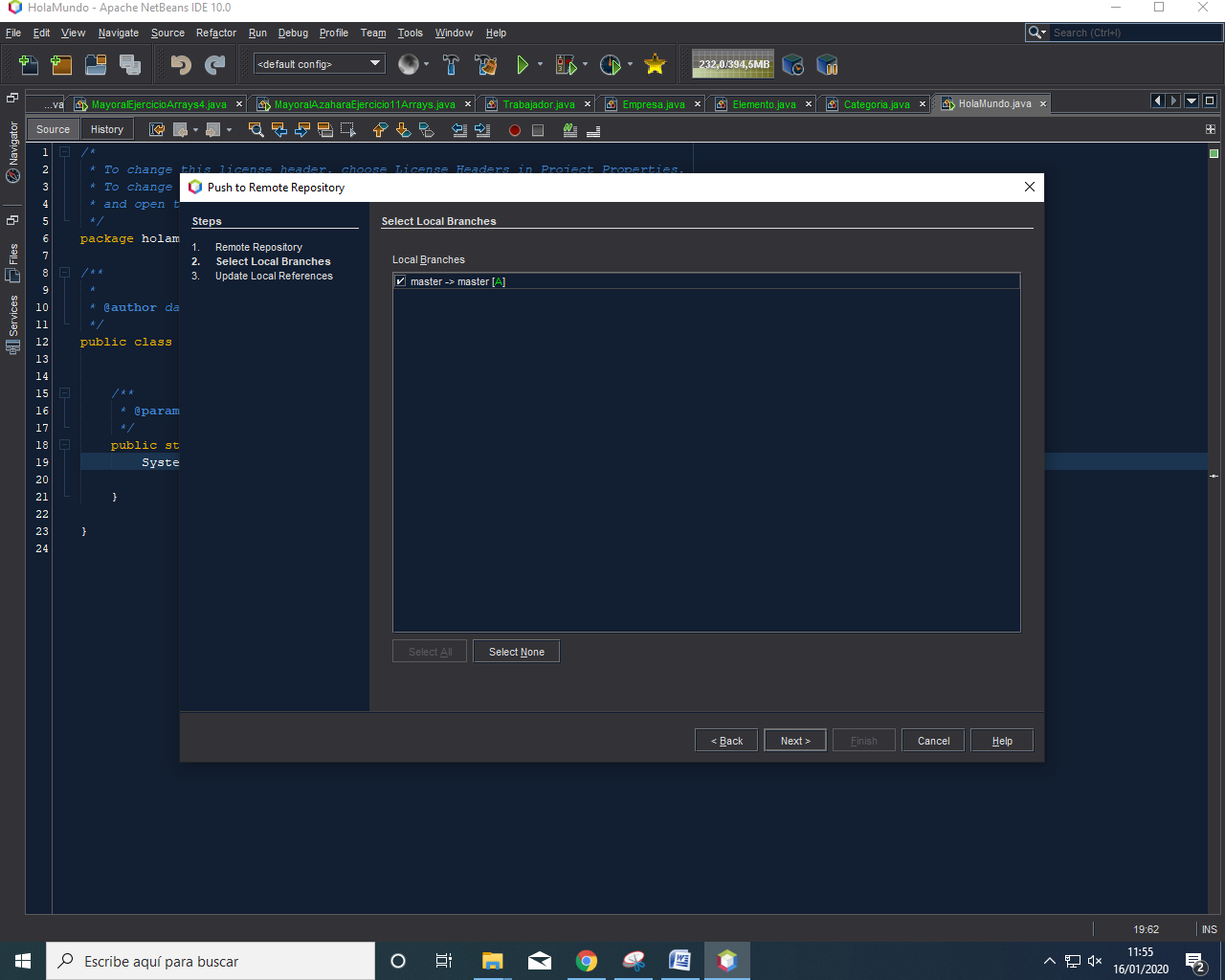


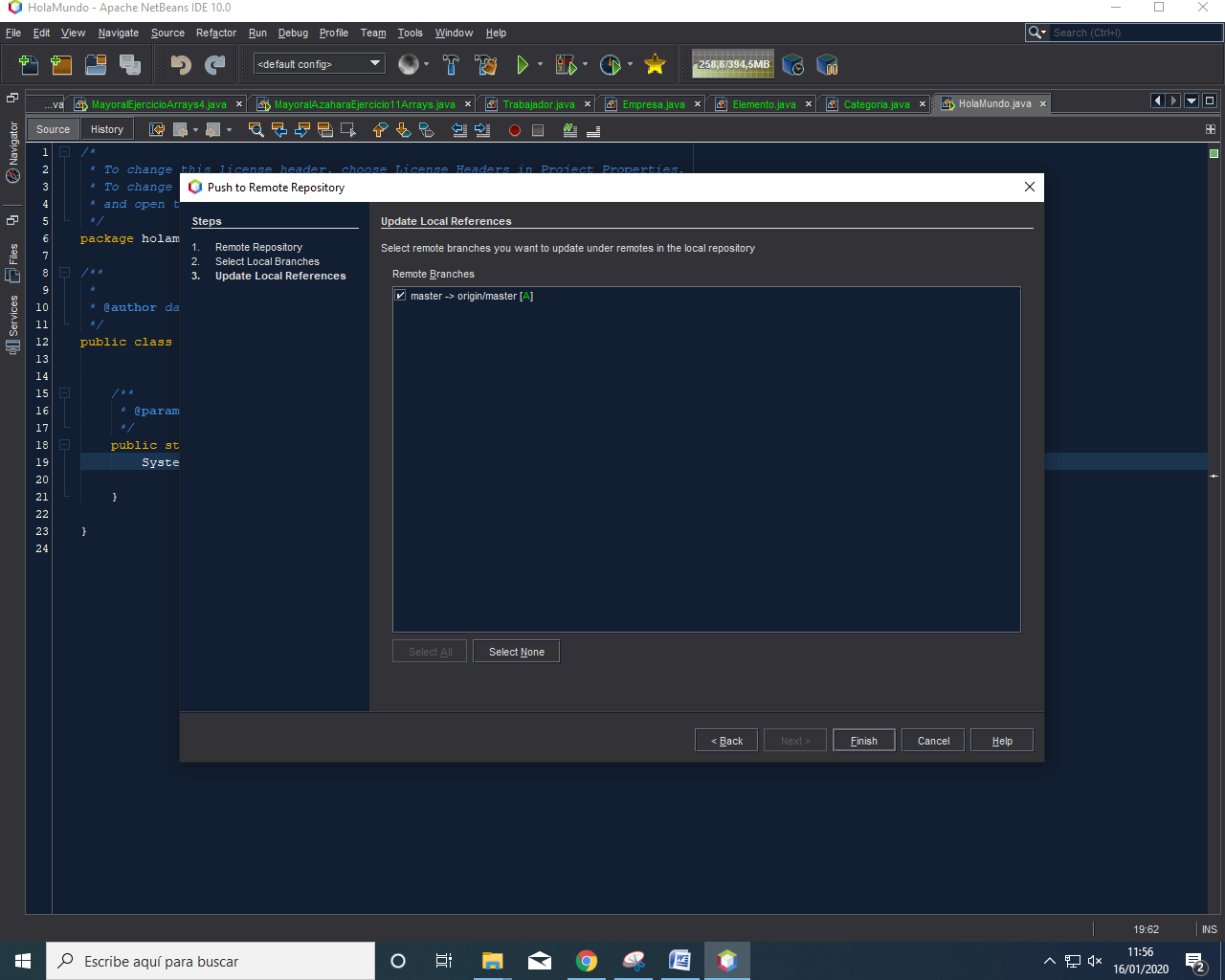
Hay que seleccionar la dirección de GitHub



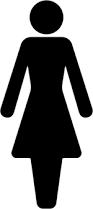
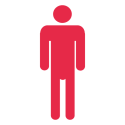
Y meter usuario y contraseña.

Añadir que quieres subir la rama Master.

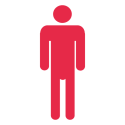




• Ahora el equipo ve en GitHub el proyecto.

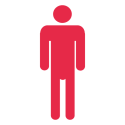
Daniel.  Azahara.  DanielDC



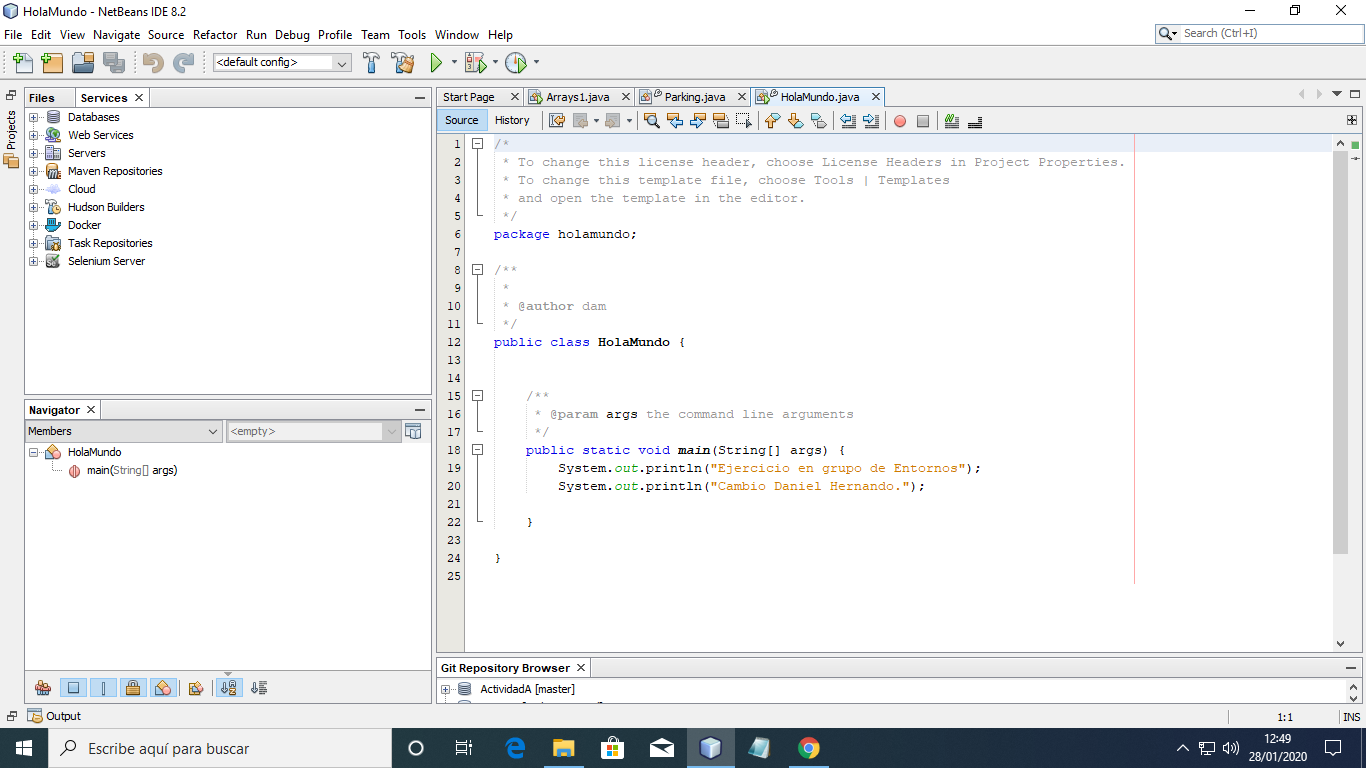
Daniel.  DanielDC

Tienen que descargar y descomprimir el proyecto. Luego lo abren desde NetBeans.

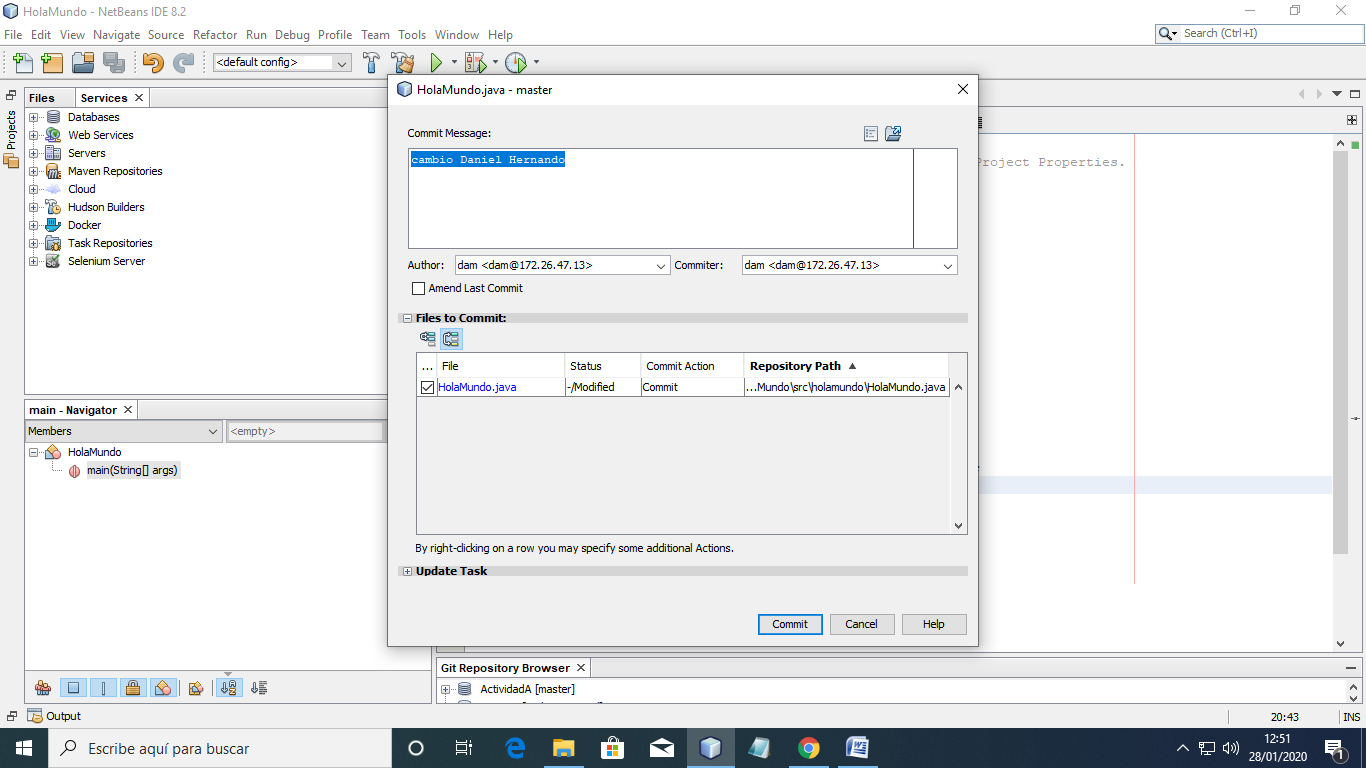
• Daniel y DanielDC modifica el archivo HolaMundo.java de nuevo y lo vuelve a subir al repositorio remoto.

Daniel.  DanielDC

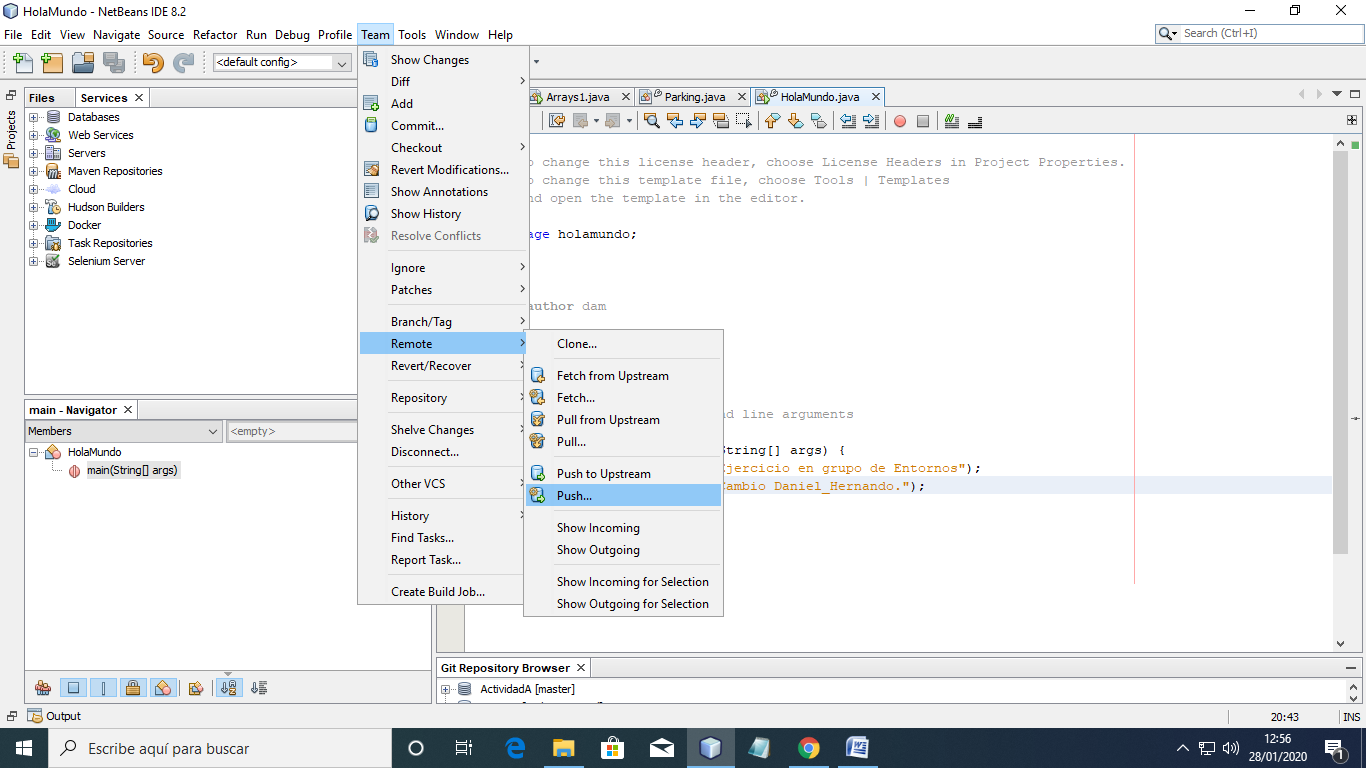
Daniel hace un cambio en el proyecto “Holamundo”



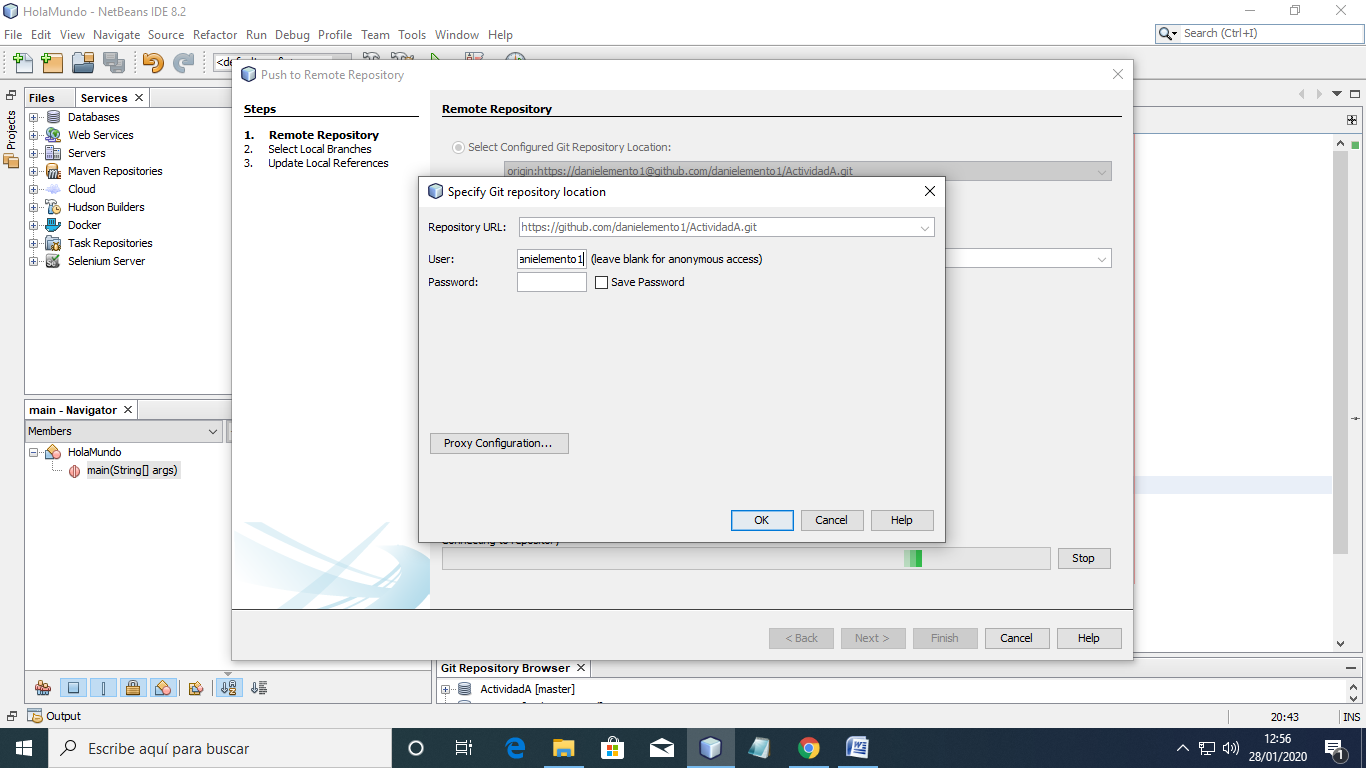
Después de hacer los cambios hace un “commit”.



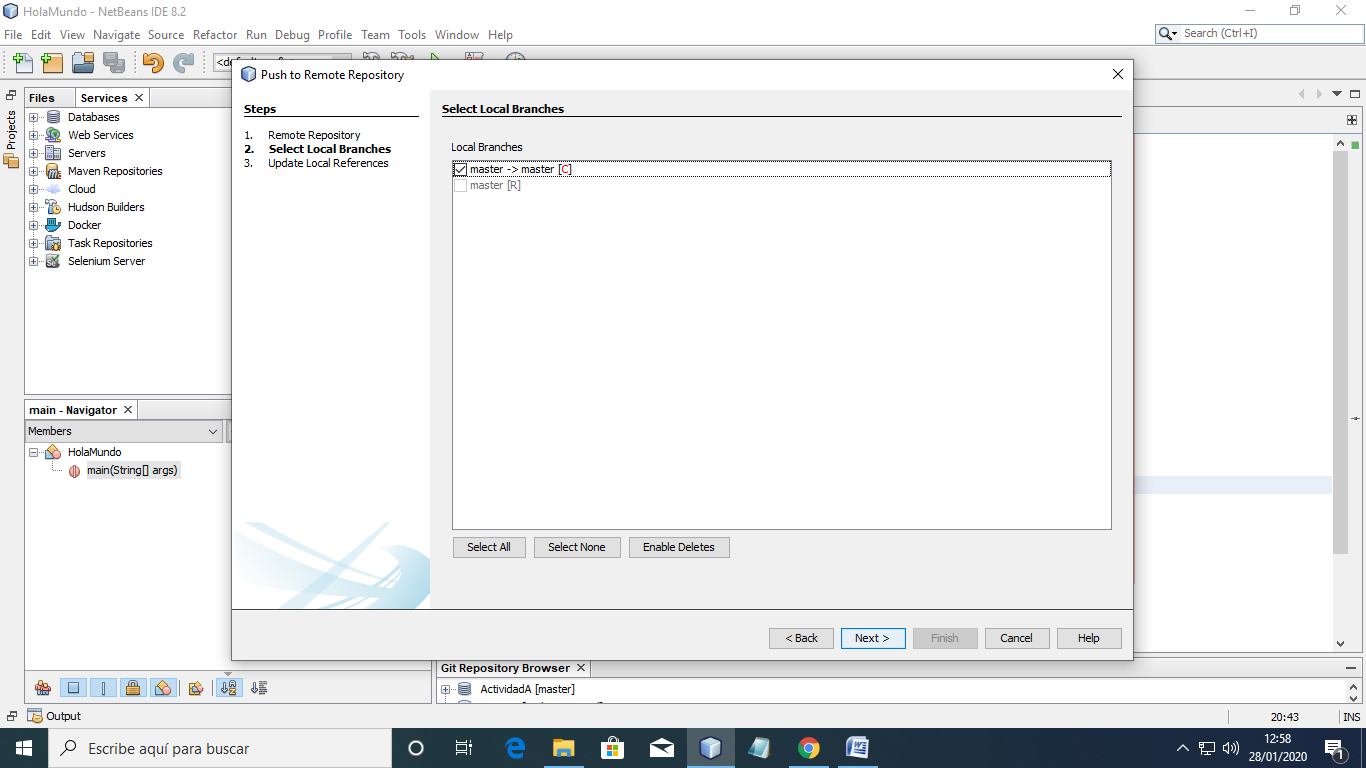
Daniel hace un “push” para subir los cambios hechos al repositorio remoto.



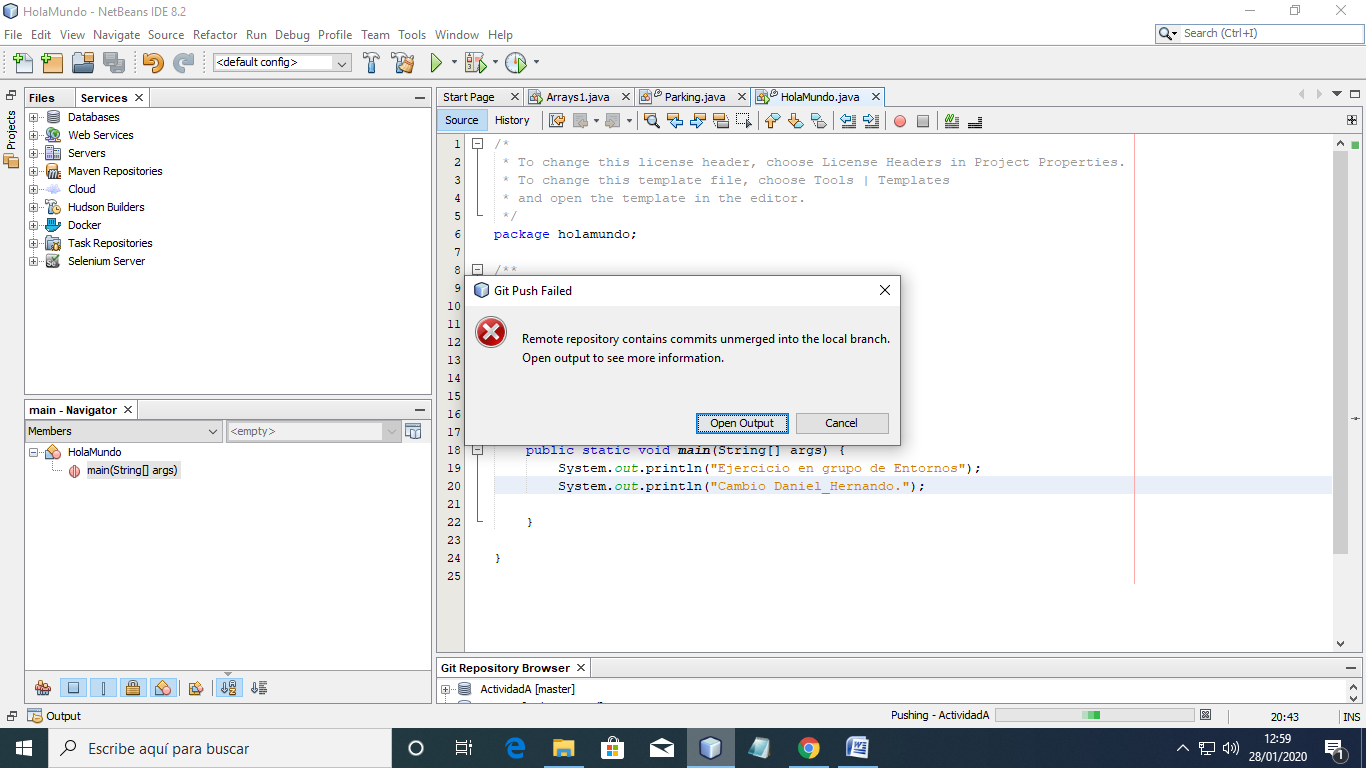
Daniel introduce la contraseña del repositorio para poder guardar los cambios



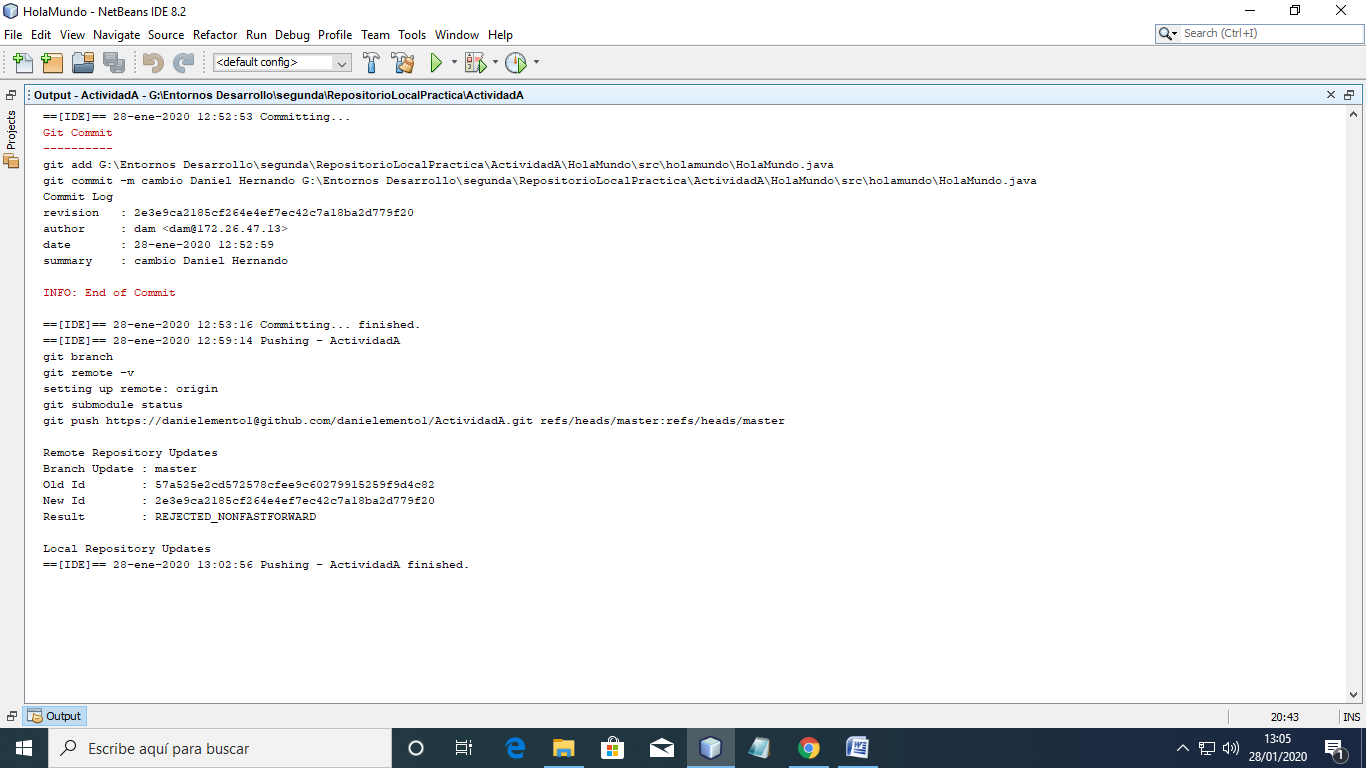
Daniel selecciona la rama maestra y pulsa siguiente hasta finalizar.



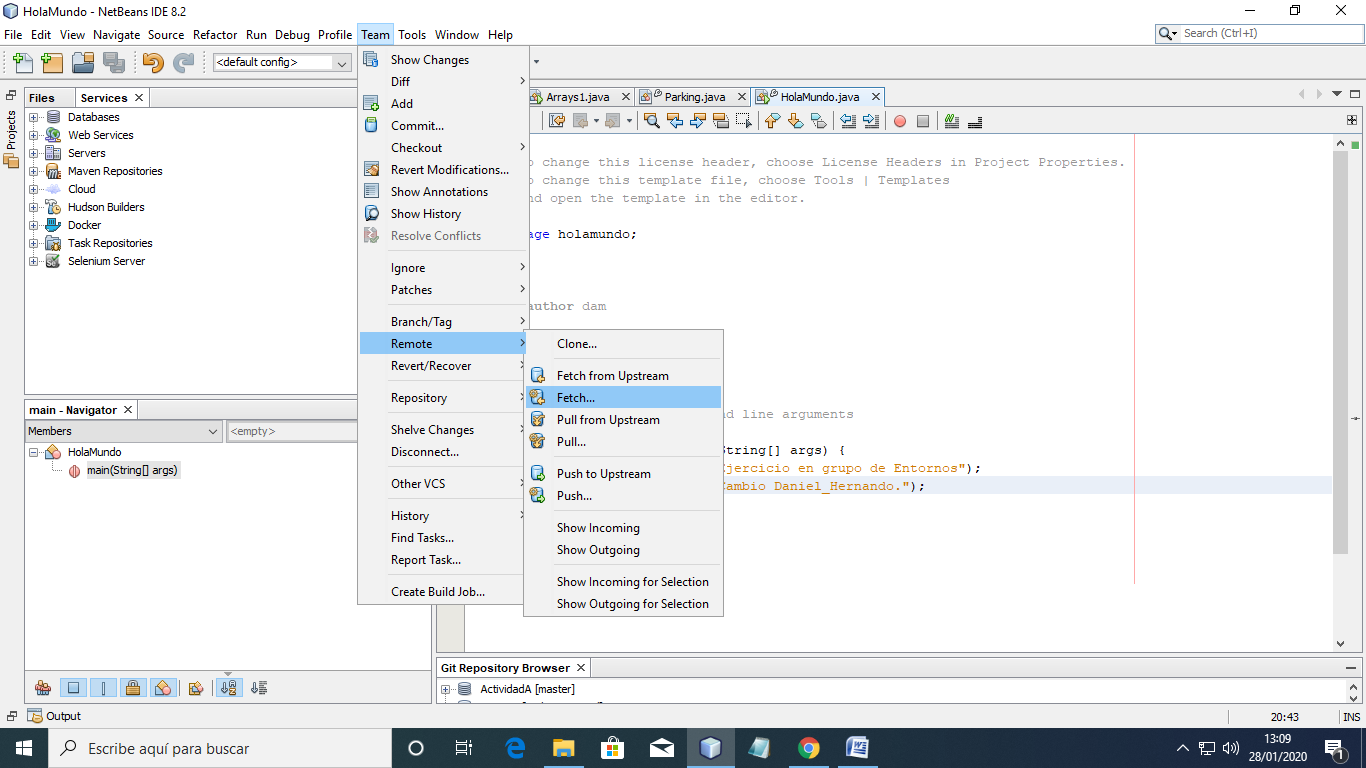
A Daniel le aparece un error que dice que hay commits que no se han fusionado.



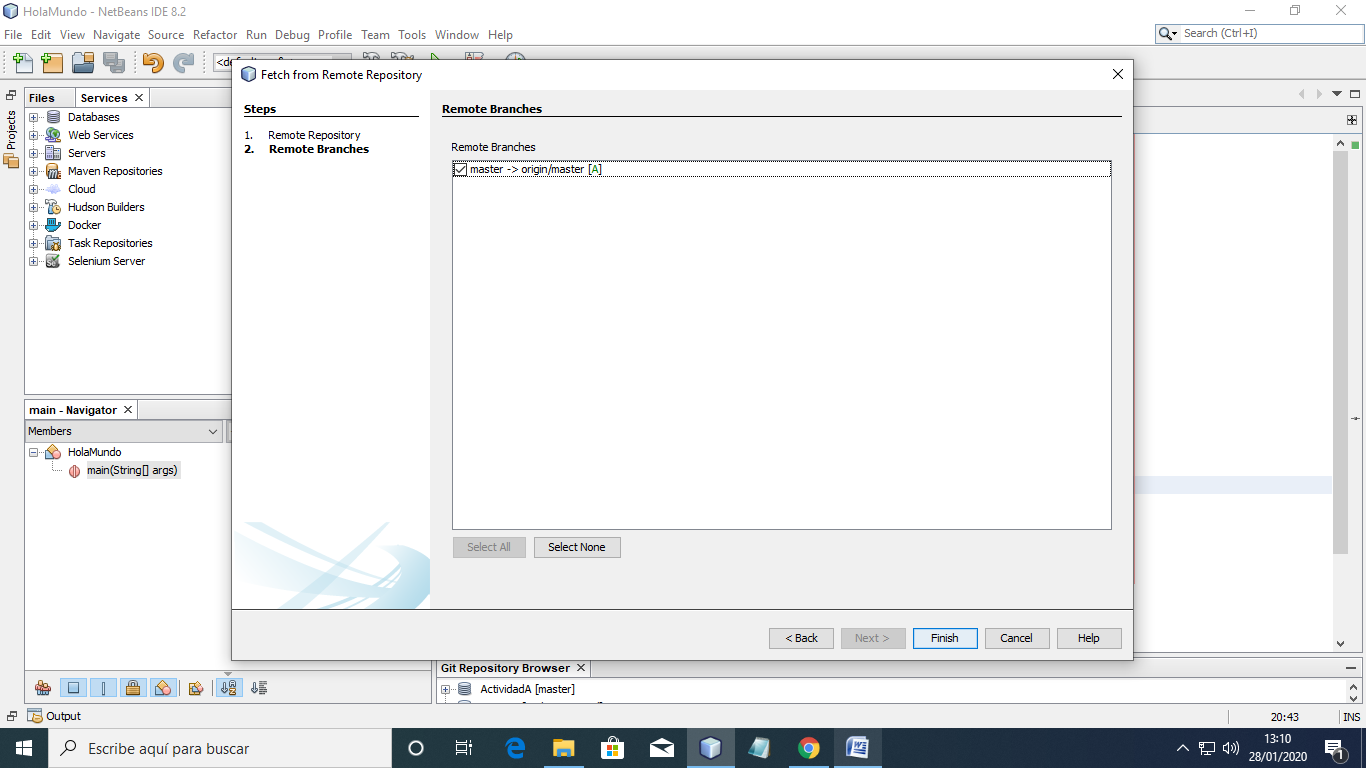
Abre el output como se indica para ver lo que ocurre.



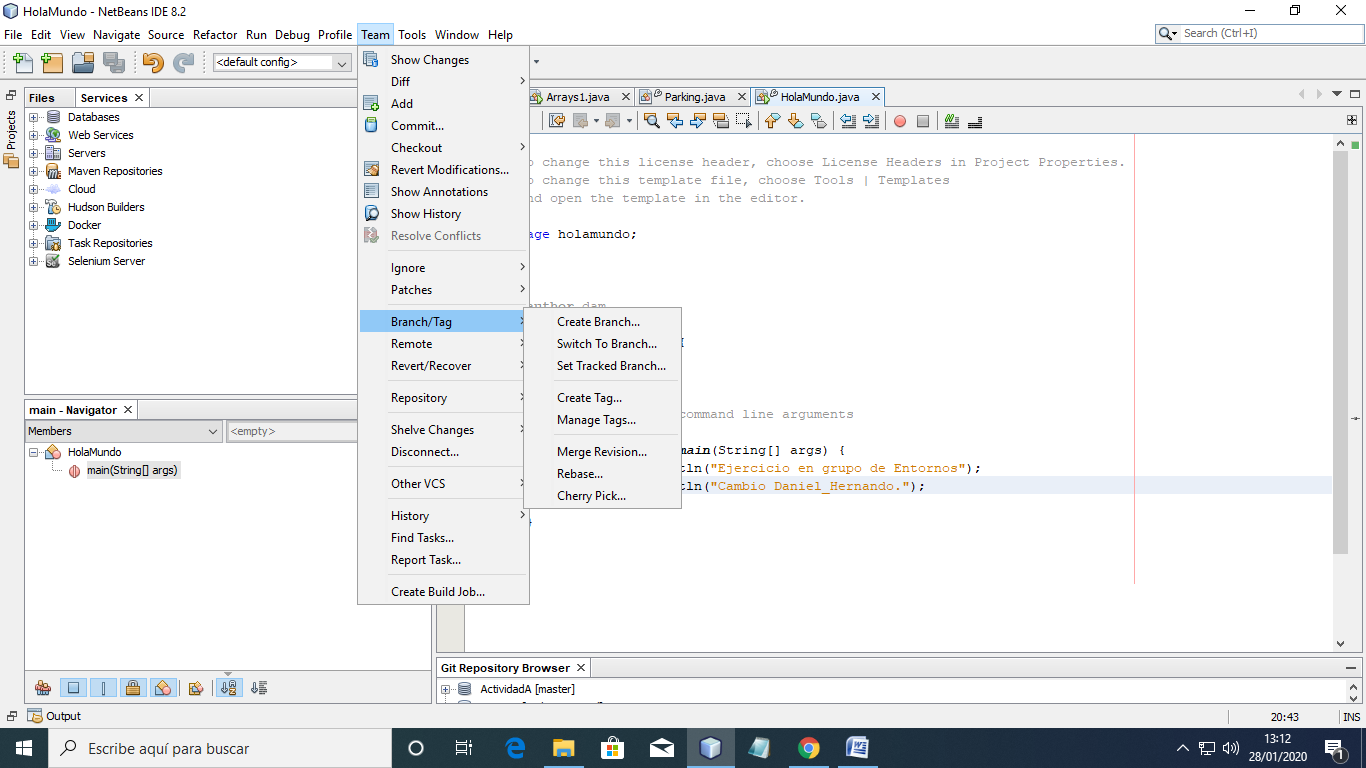
El fallo ocurrido es porque Azahara hizo un push sobre la rama en la que Daniel está trabajando y cuando Daniel fue a subir sus cambios entra en conflicto con los cambios hechos por Azahara. Para resolver este problema vamos a TEAM – REMOTE –FETCH

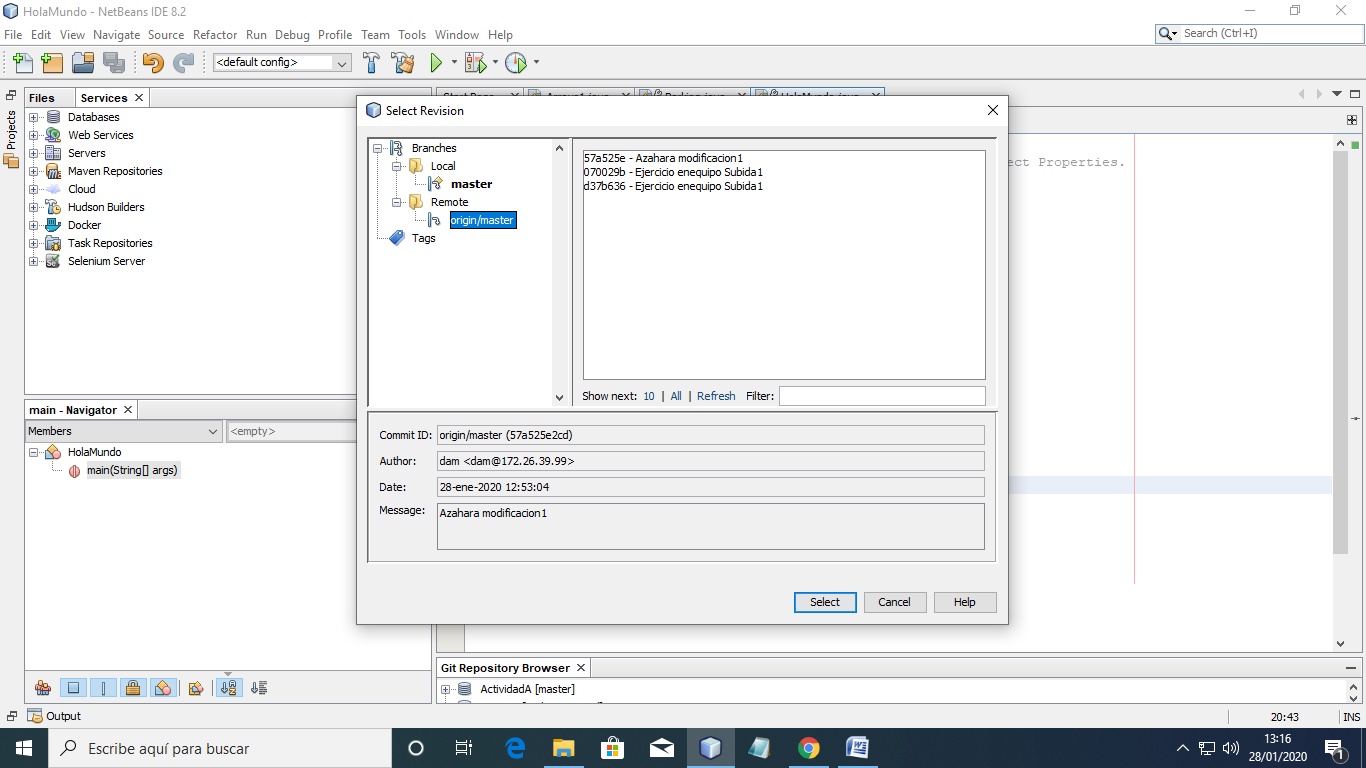


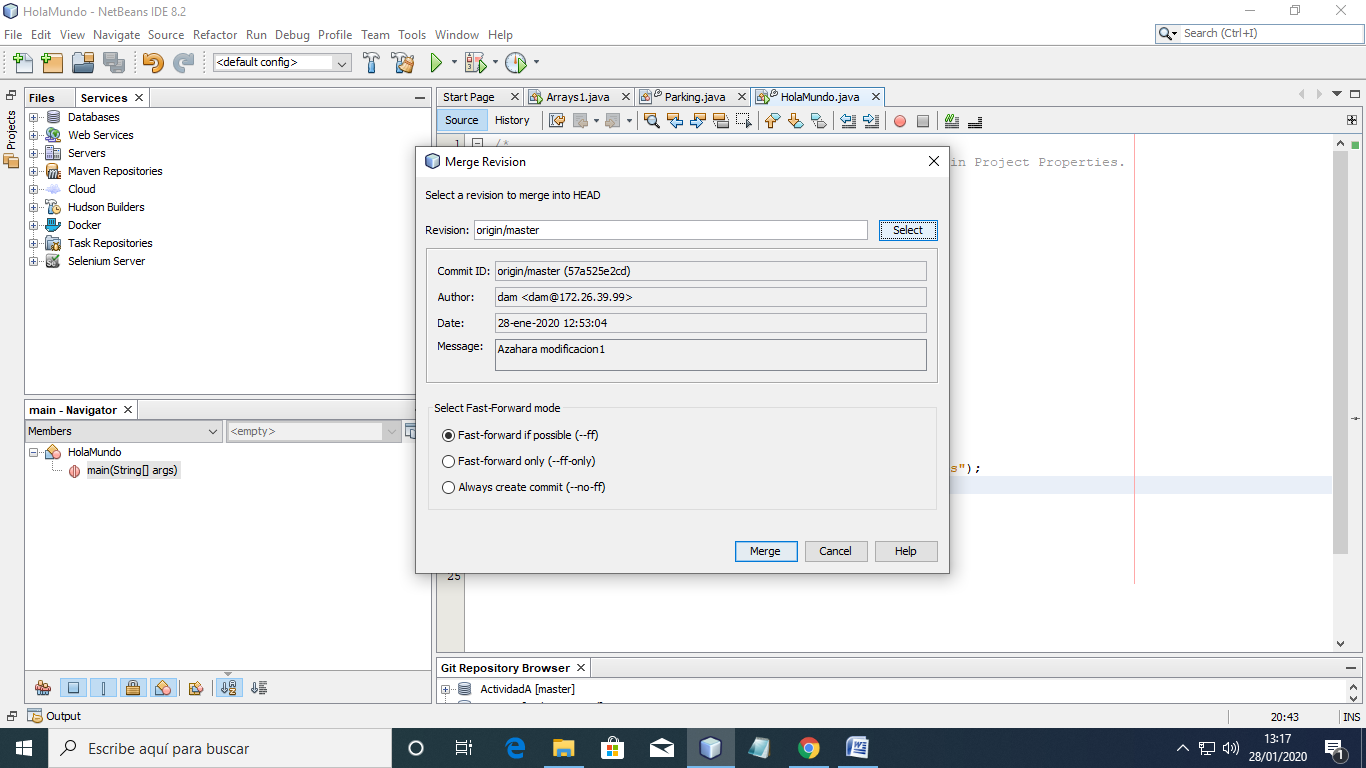
Daniel selecciona la rama master y pulsa “finish”.

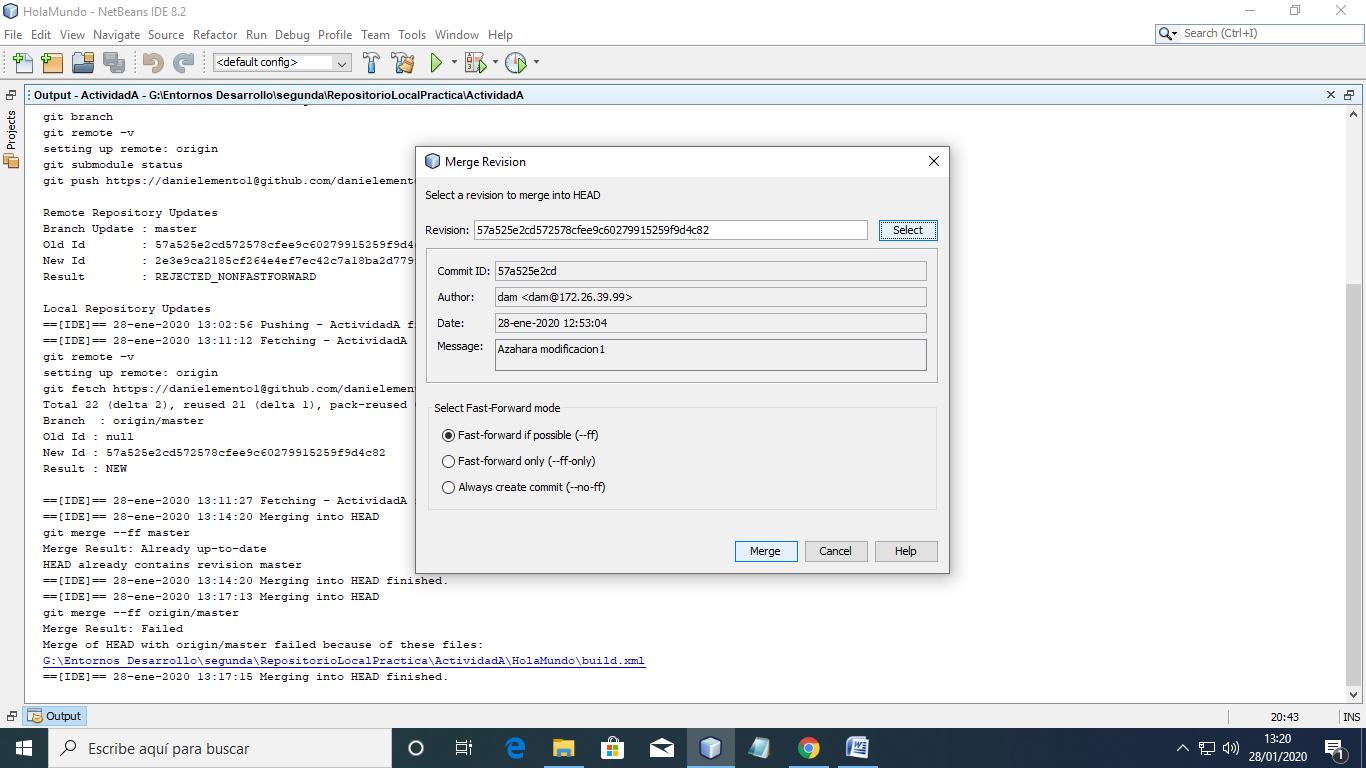


Ahora vamos a Team, dentro de este a BRANCH, y luego a MERGE REVISION.

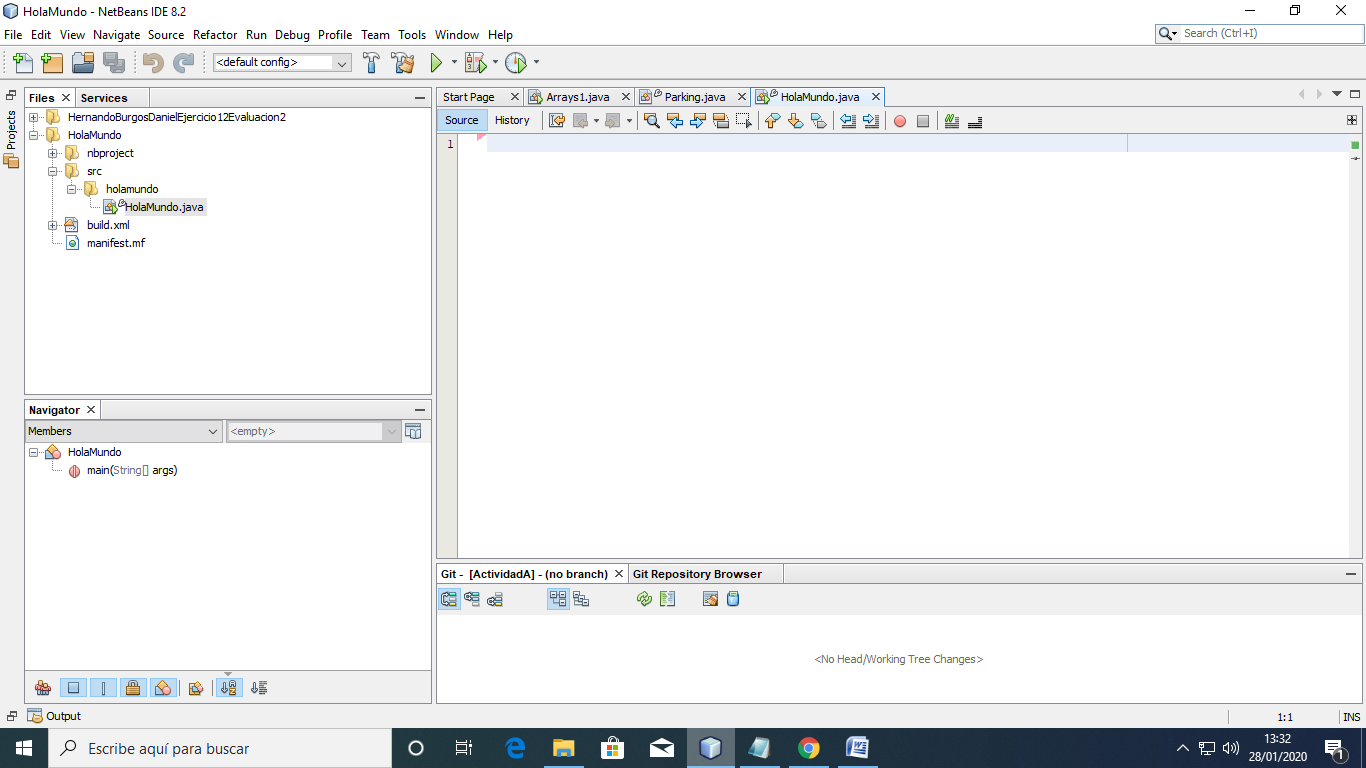




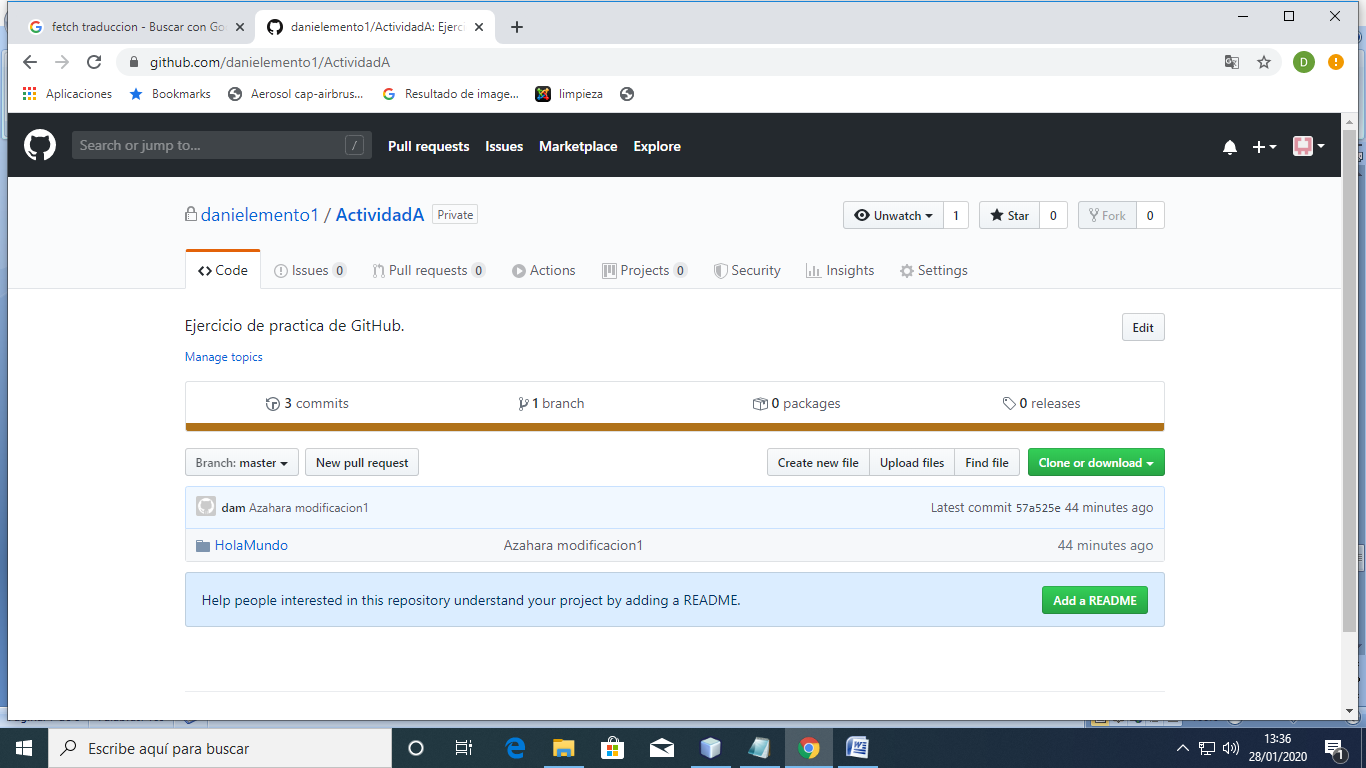




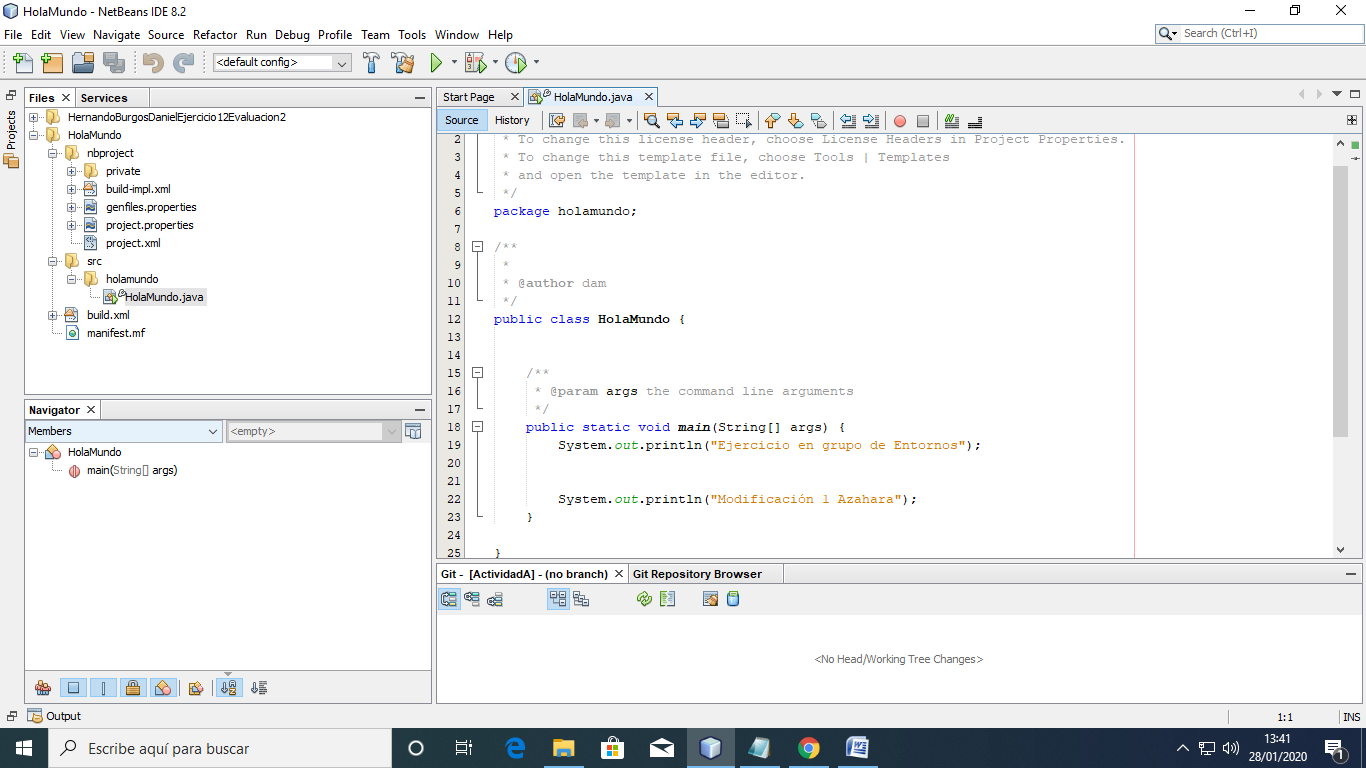
Después de hacer el merge nos da conflicto ya que hay 3 colaboradores trabajando sobre la misma rama y nos deja el proyecto en blanco



Para solucionar este problema los colaboradores van a clonar el proyecto desde el repositorio en github para que todos los colaboradores tengan el mismo proyecto de nuevo. En futuras ocasiones antes de hacer un push se hará un pull para recoger el proyecto como está actualmente.

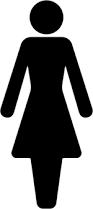


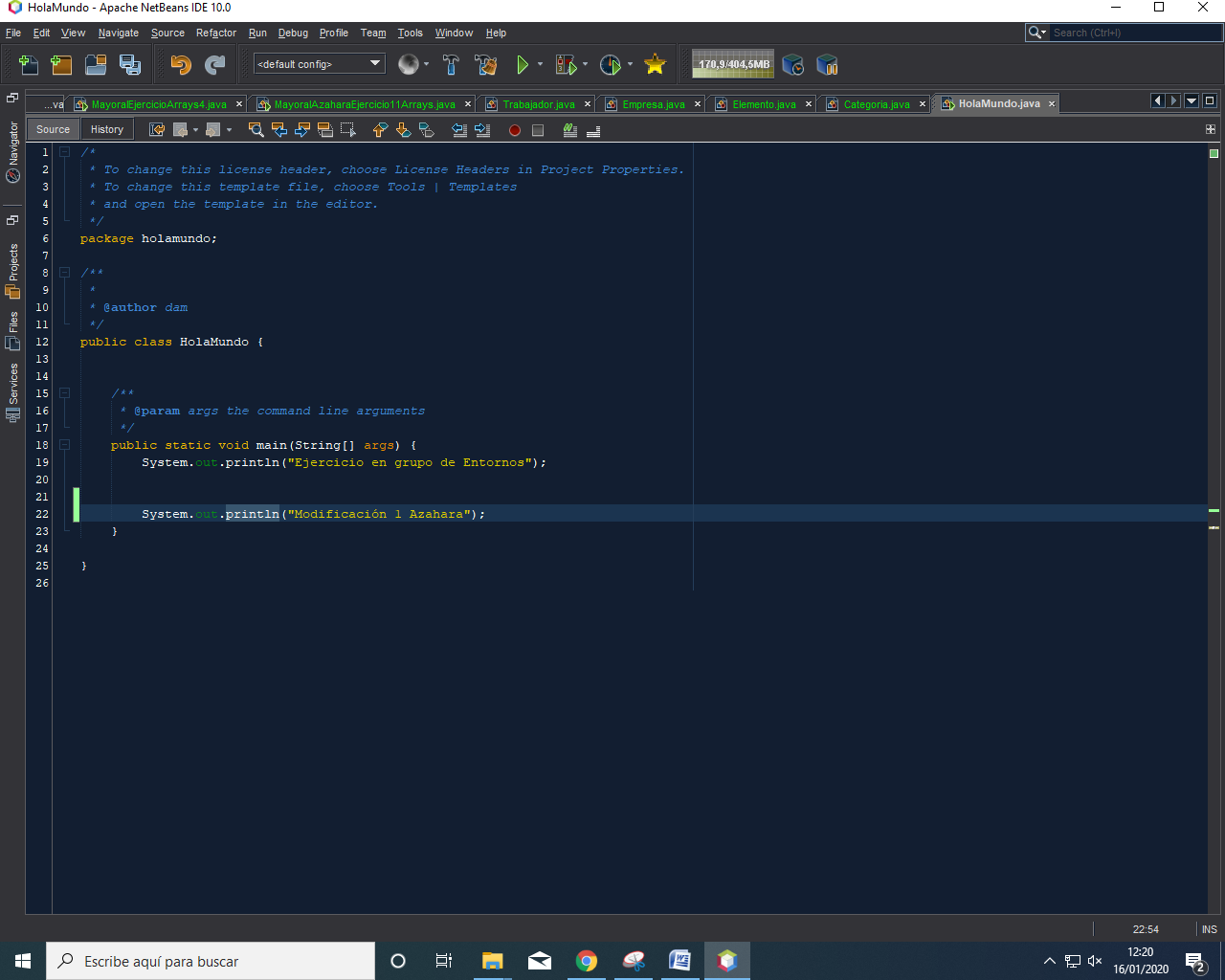
Una vez clonado lo importamos para tener el proyecto en nuestro equipo.



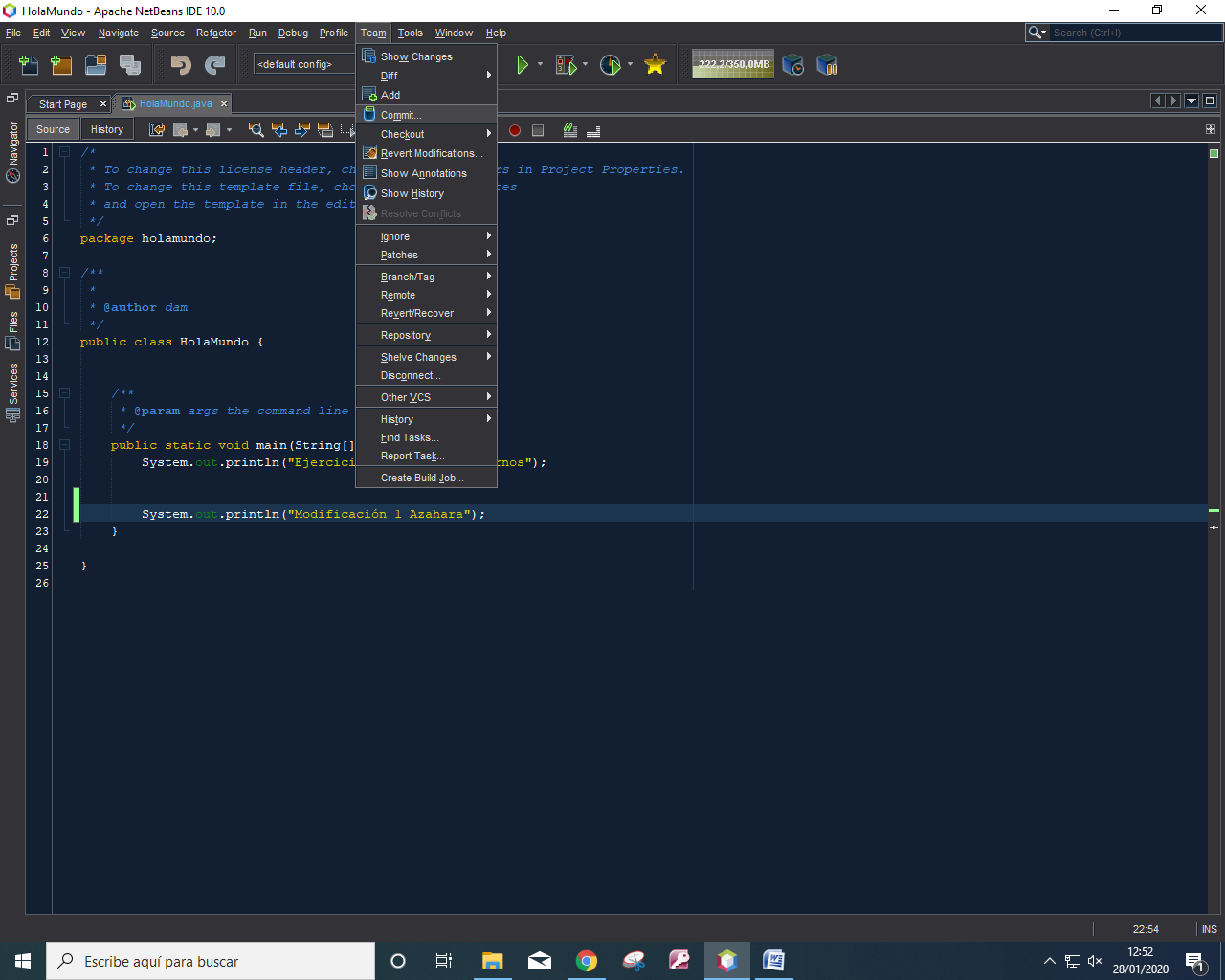
Mientras tanto, Azahara modifica el archivo HolaMundo.java que tenía en su repositorio local añadiendo “Modificación 1 Azahara”.

Explica que ocurre.

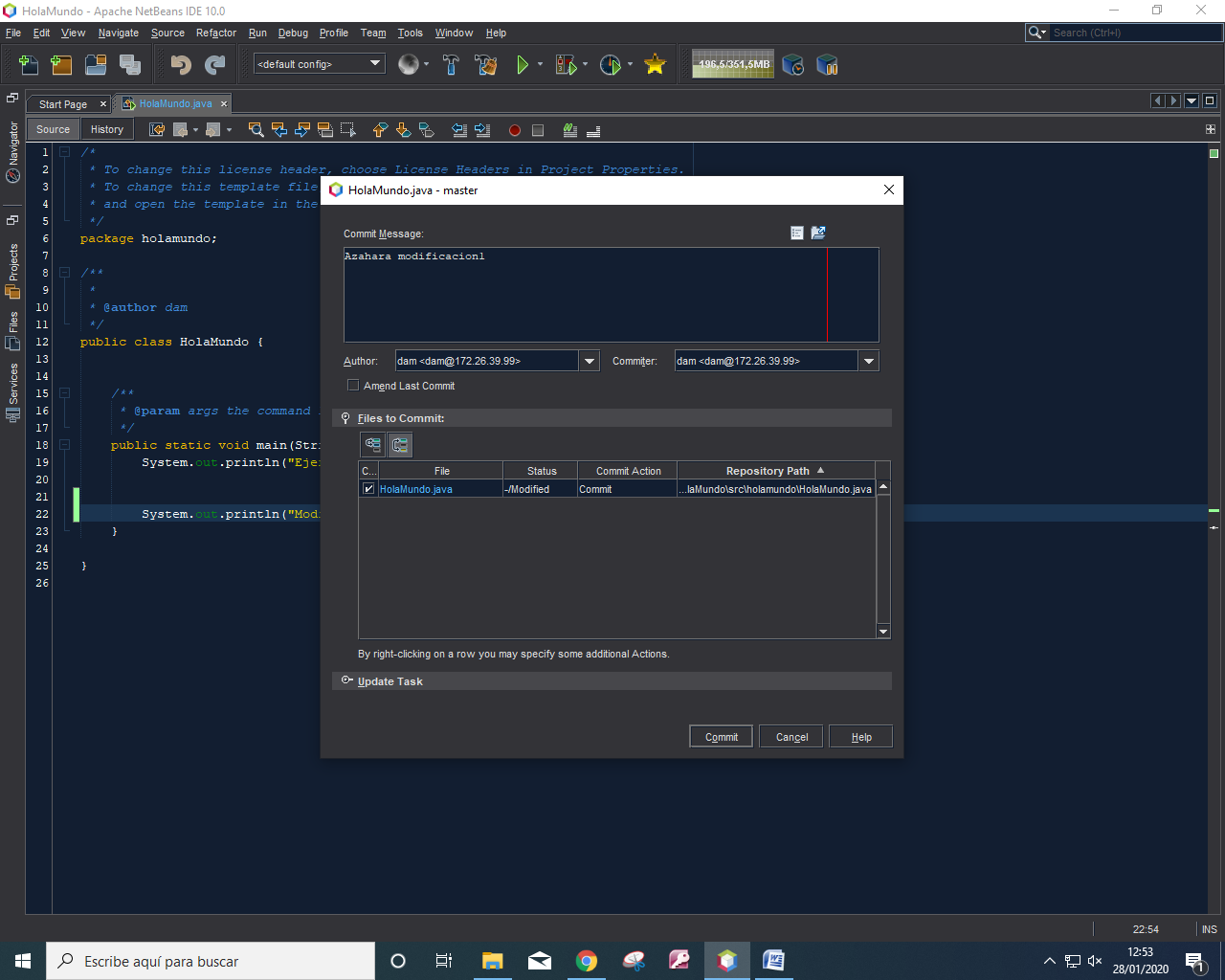
Azahara.



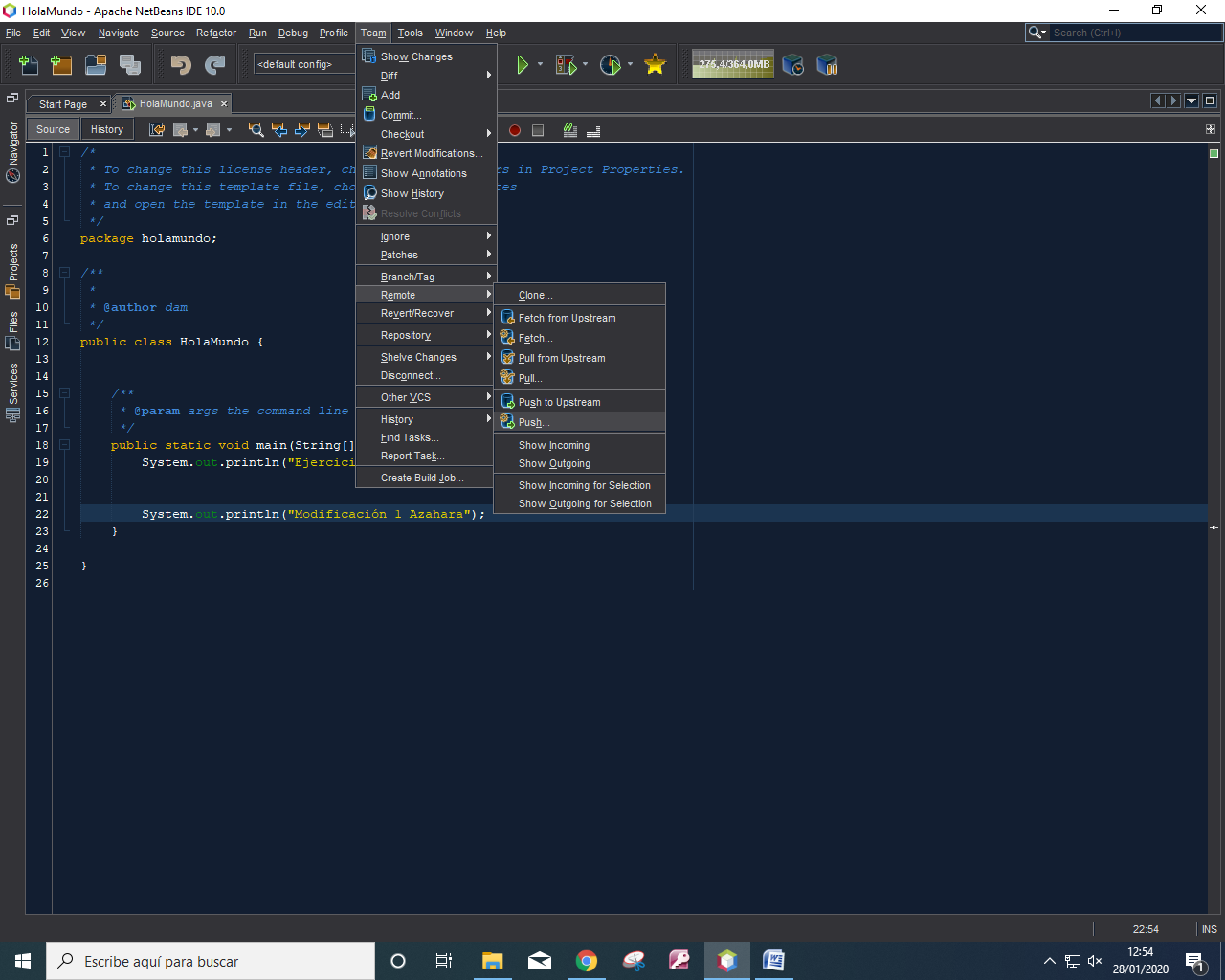
Azahara ha creado la línea “Modificacion 1 Azahara” y realiza un Comit:

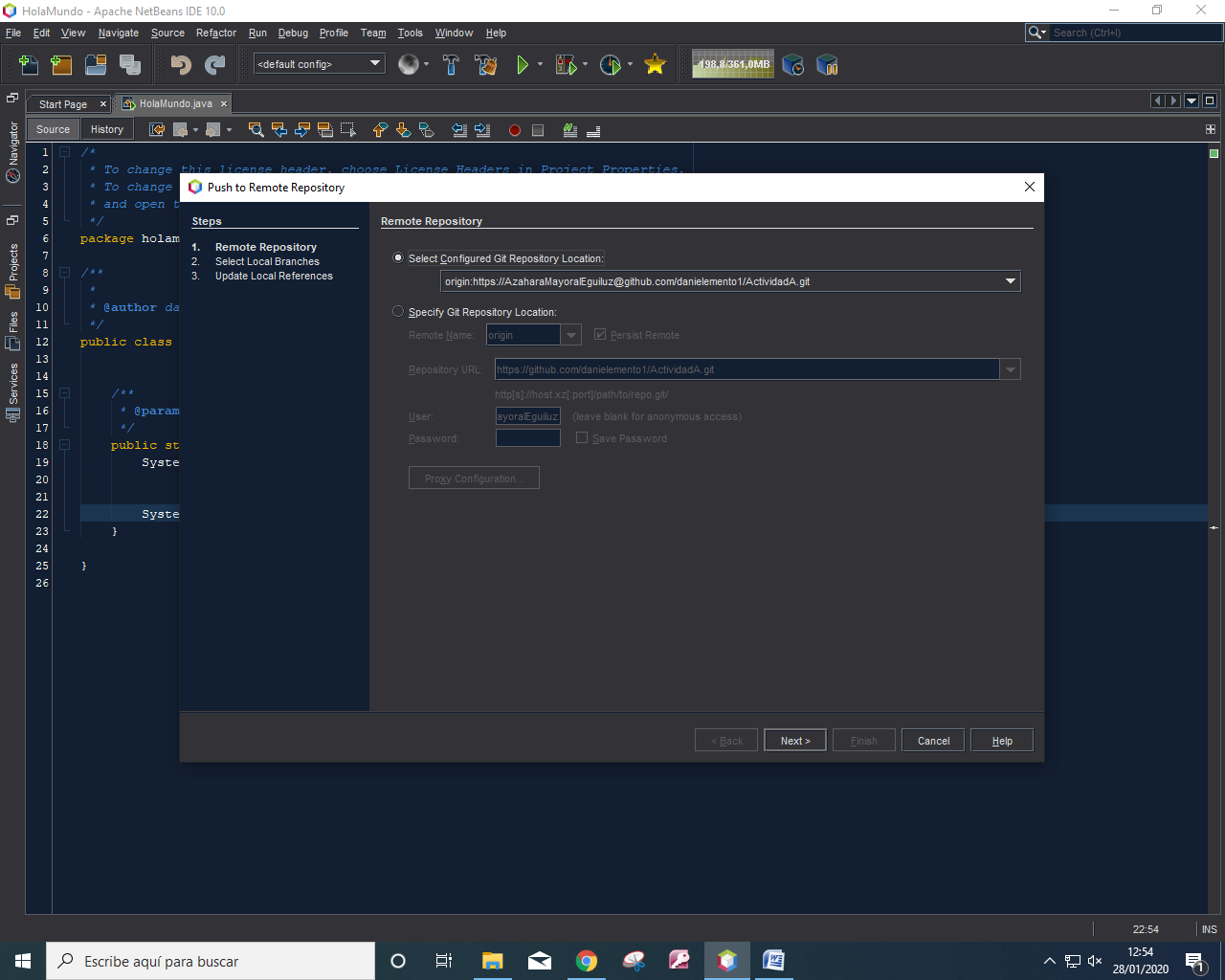
:

Le asigna el nombre de Azahara Modificacion1.

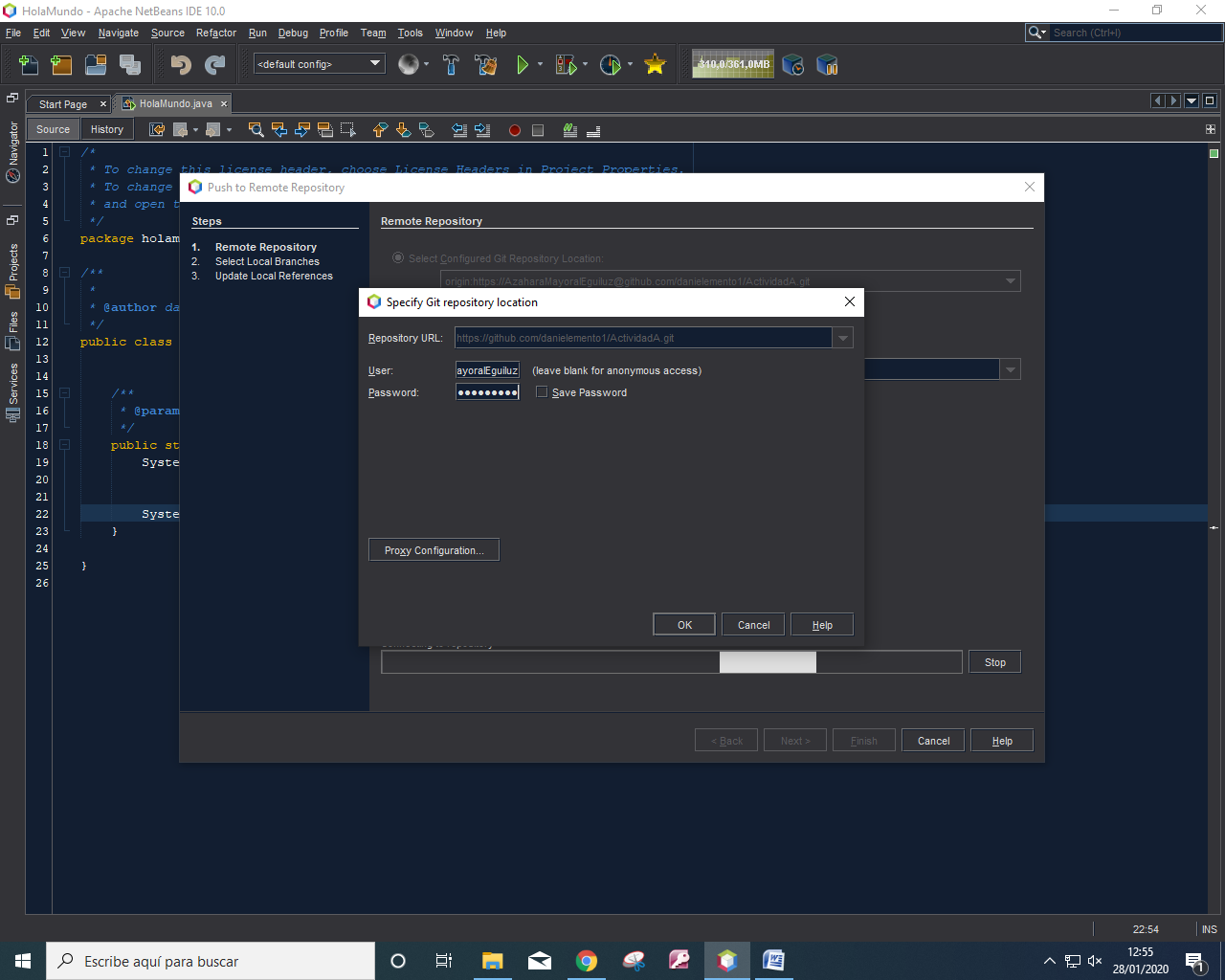


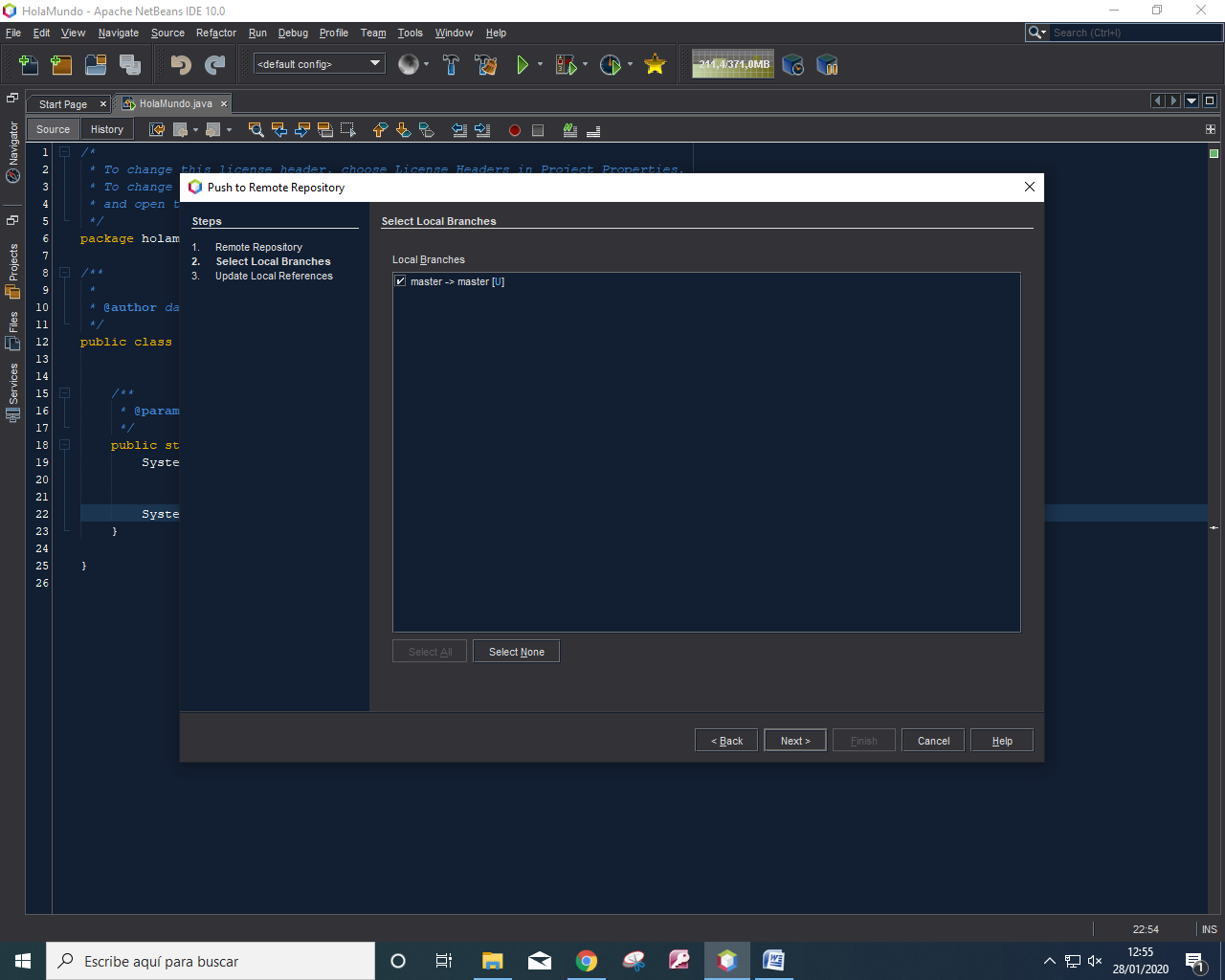
Tras el Comit, realiza un Push:





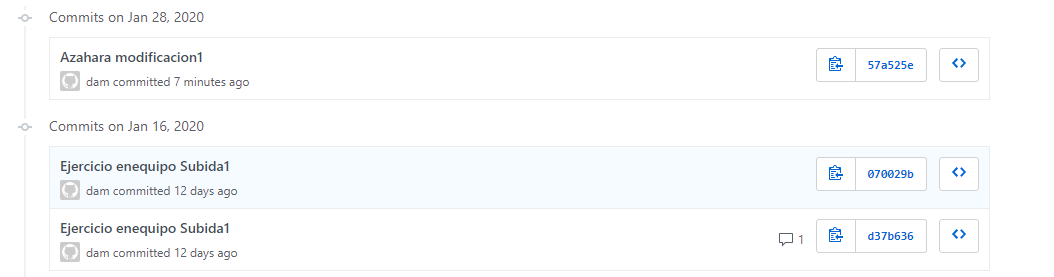
Requiere meter la contraseña y usuario:



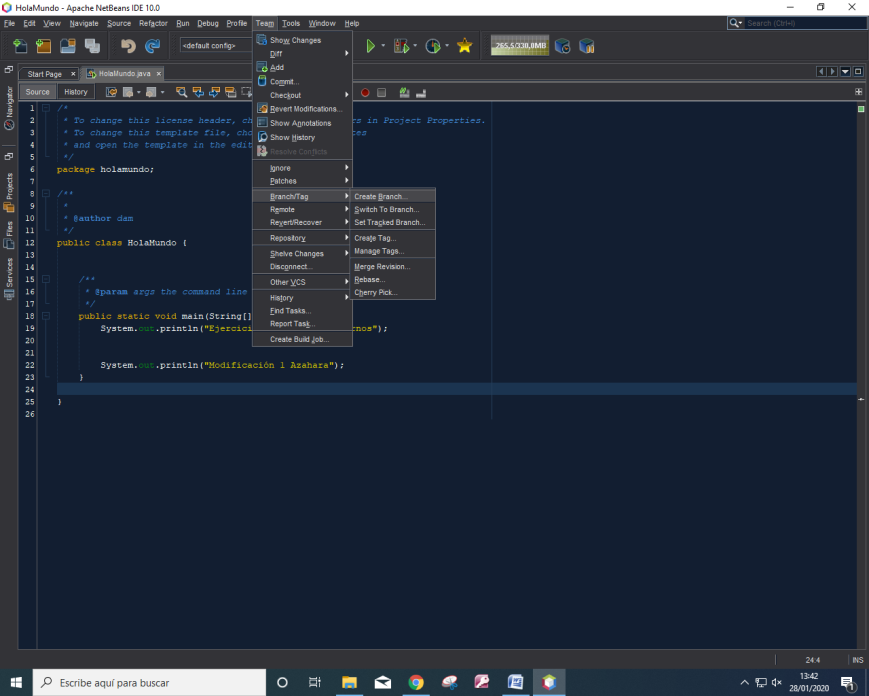


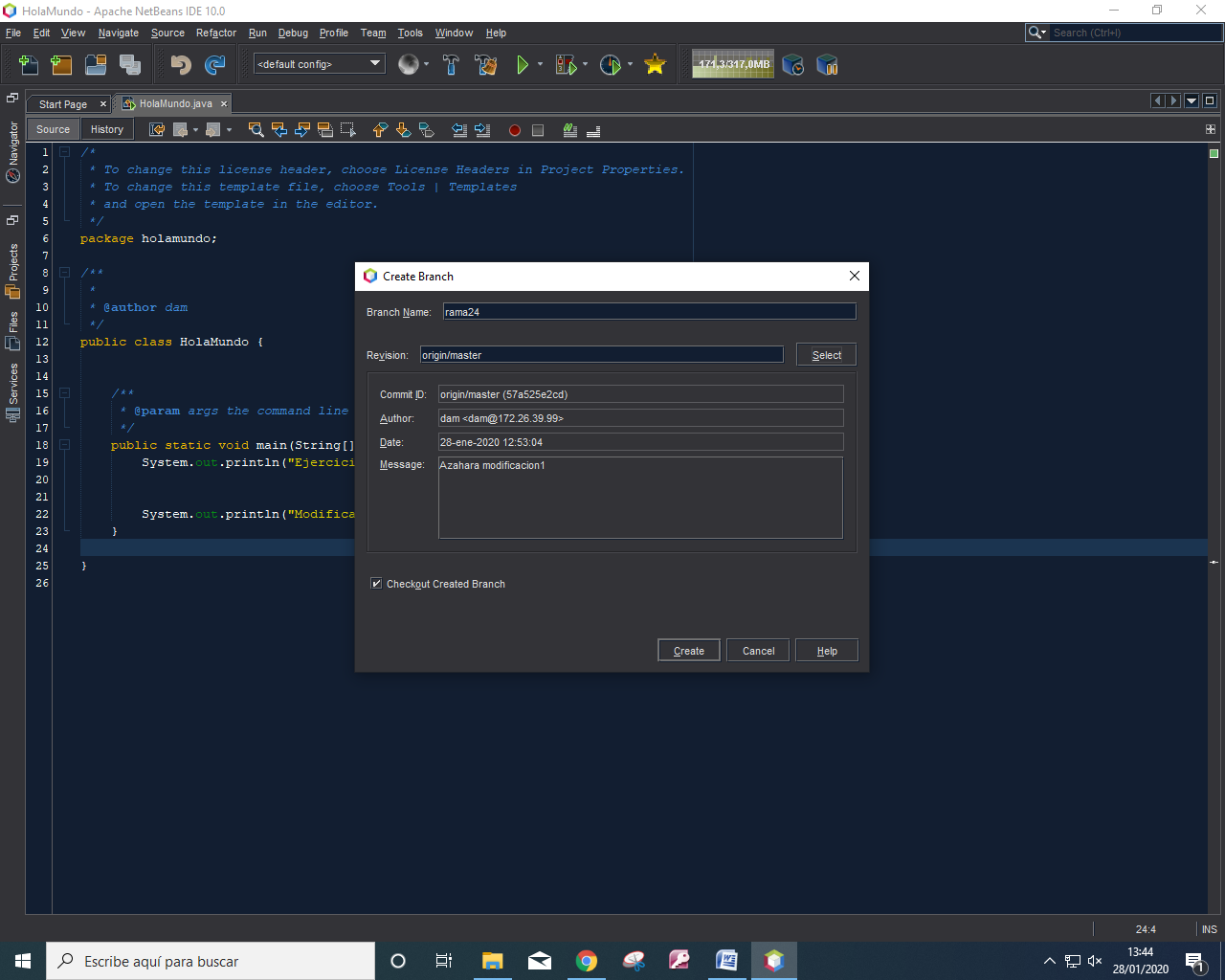


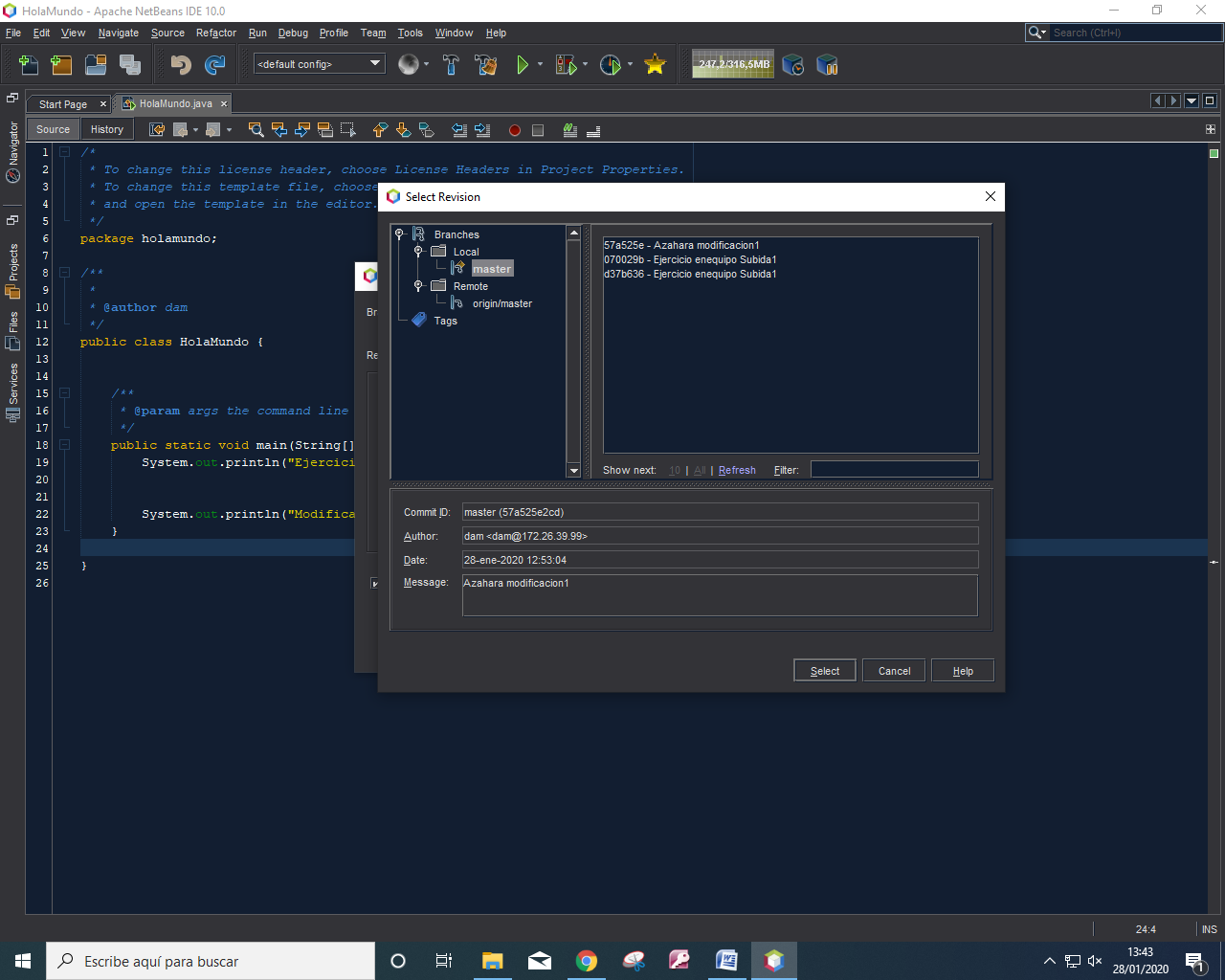
En Git se visualiza lo siguiente:



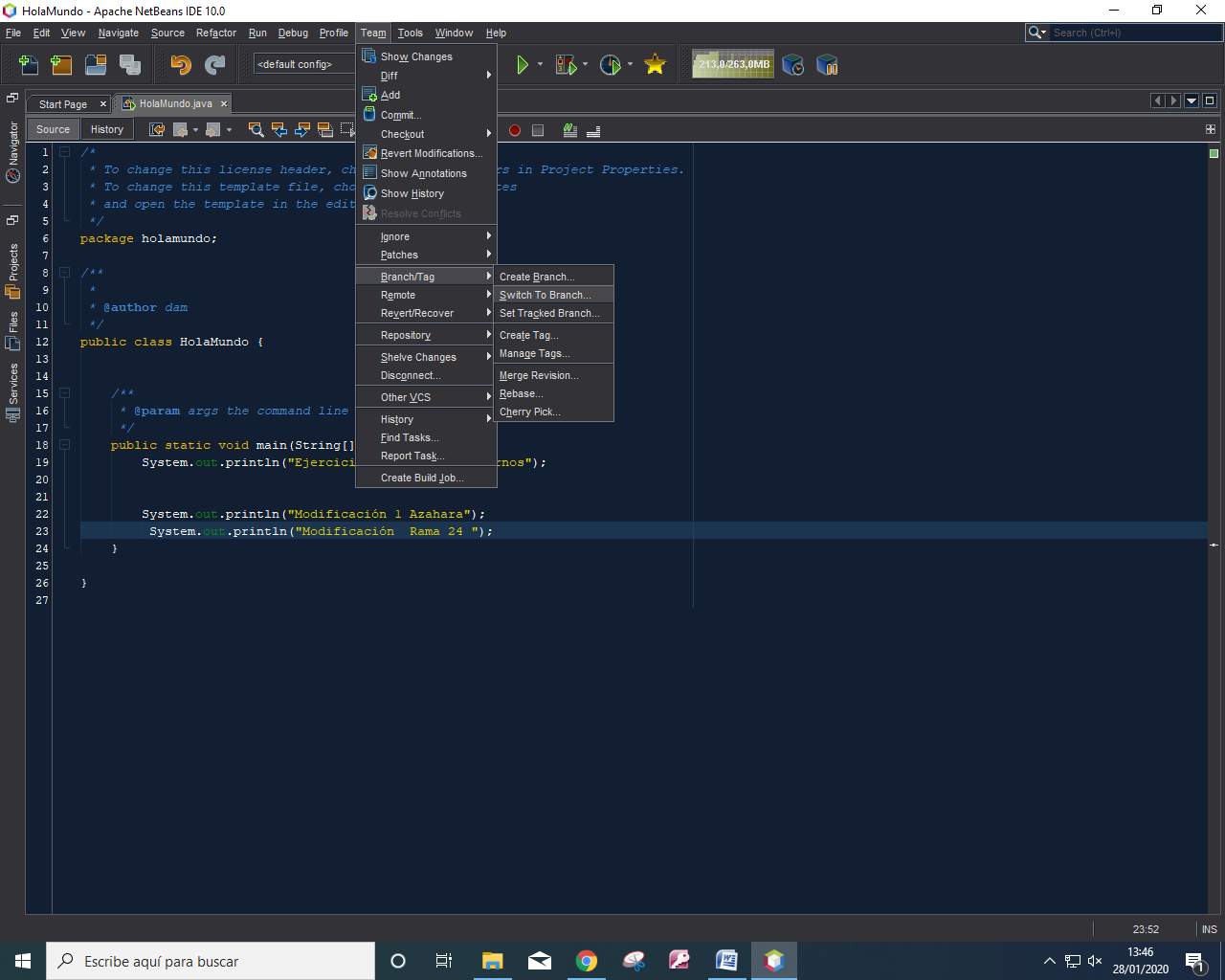
Hay un problema colectivo y no se puede solucionar y decide dejarlo a blanco, hay veces que no sabe cómo solucionarlo.

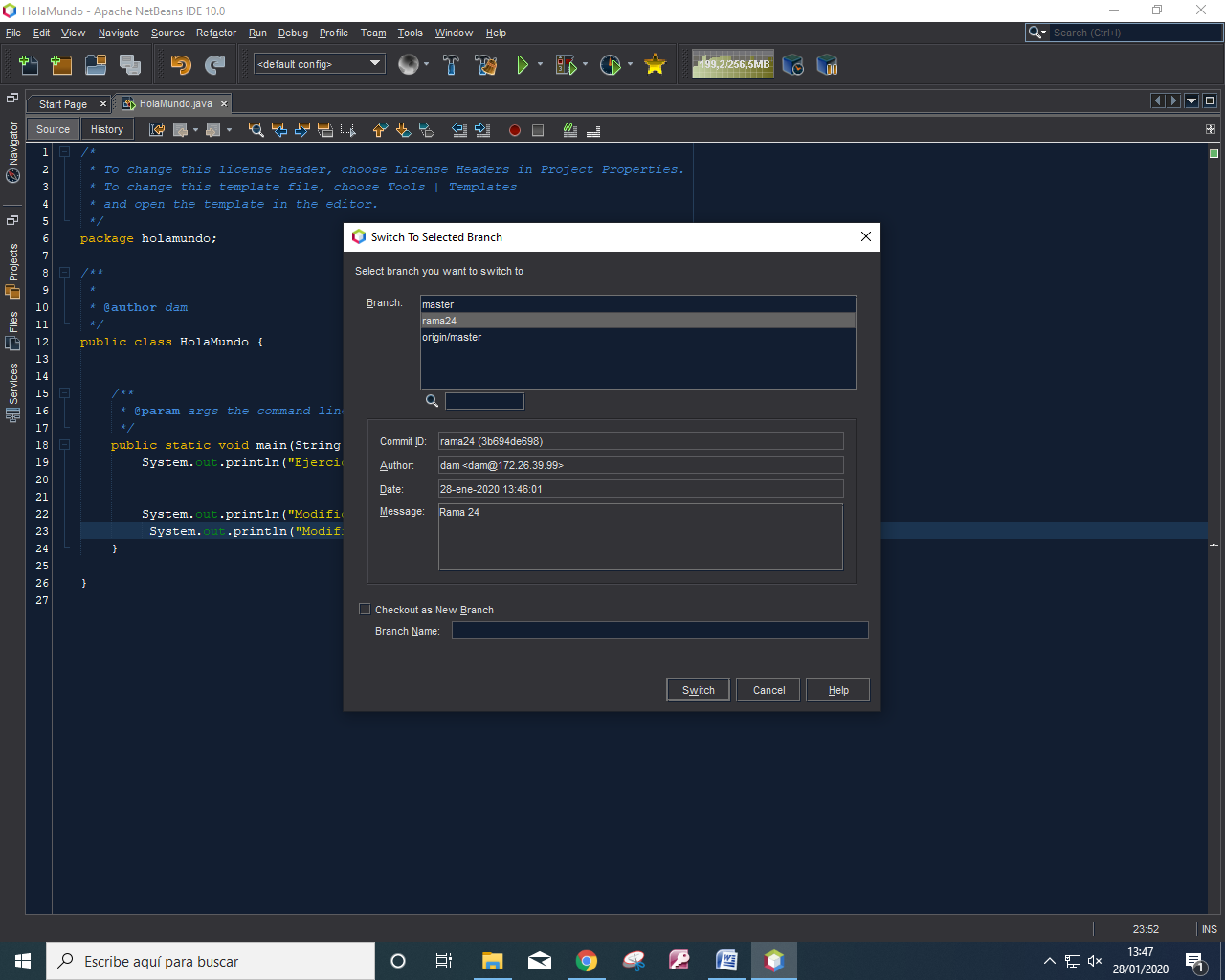
•Azahara ha estado trabajando en una rama local aparte, llamada rama24. Indica que debe de hacer para crear la rama y tenerla activa: .





Para colocarnos en la rama:

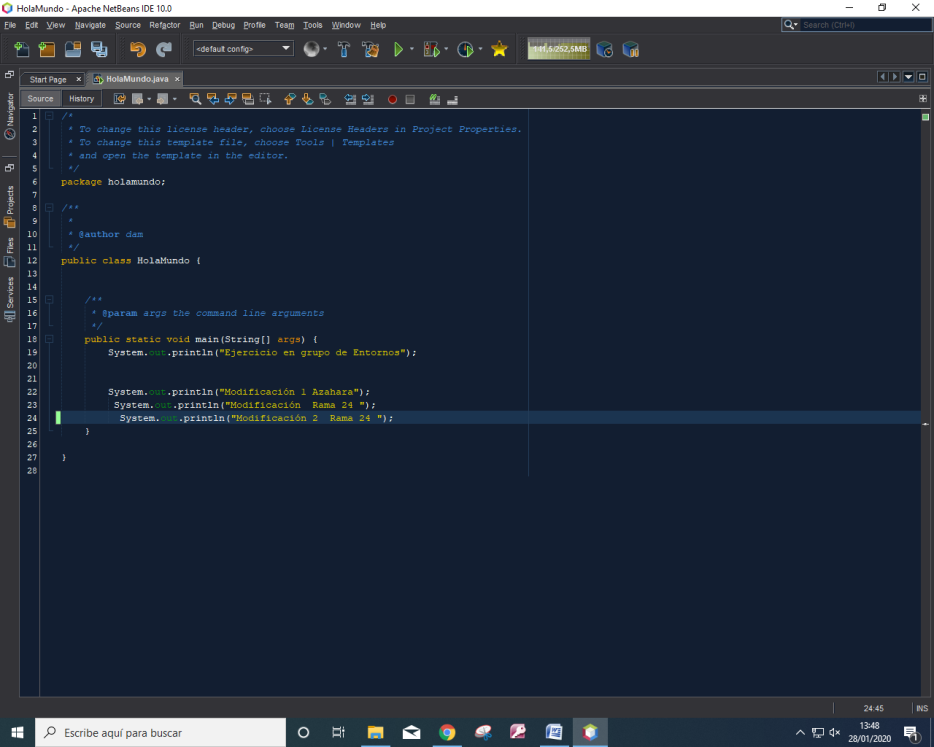




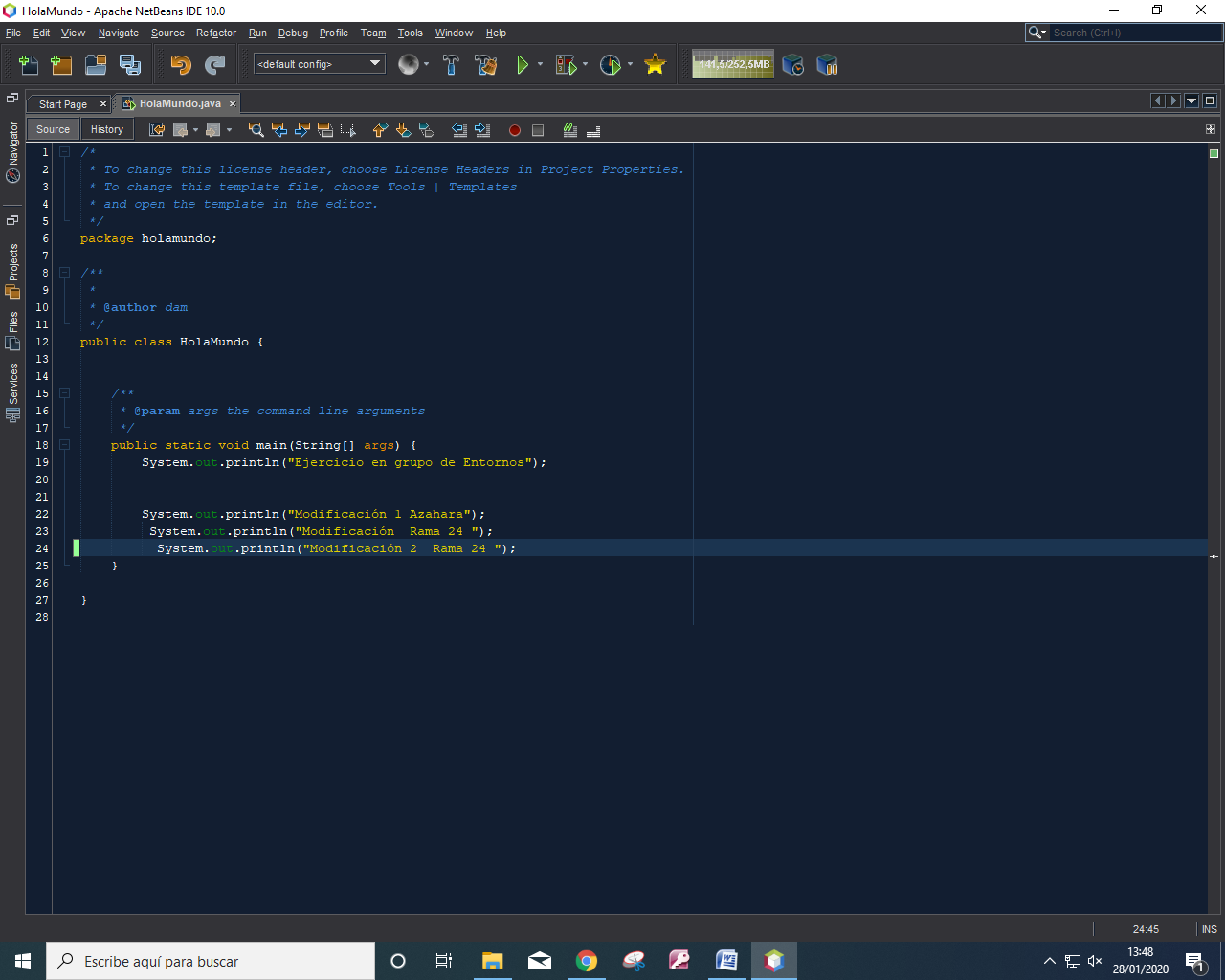
Se puede trabajar sobre las ramas o sobre la master.

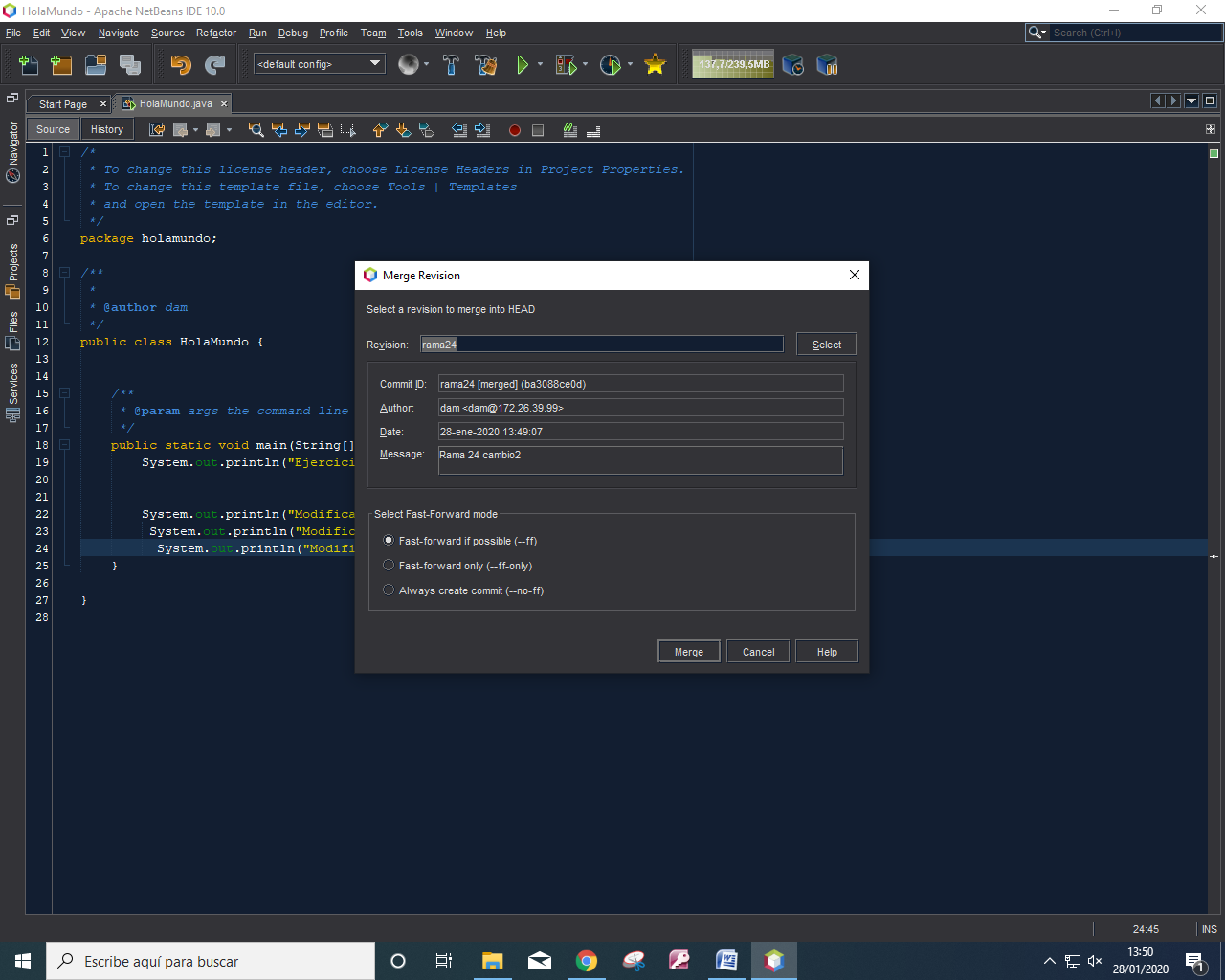
Ahora tiene dos ramas en su repositorio local: master y rama24 .

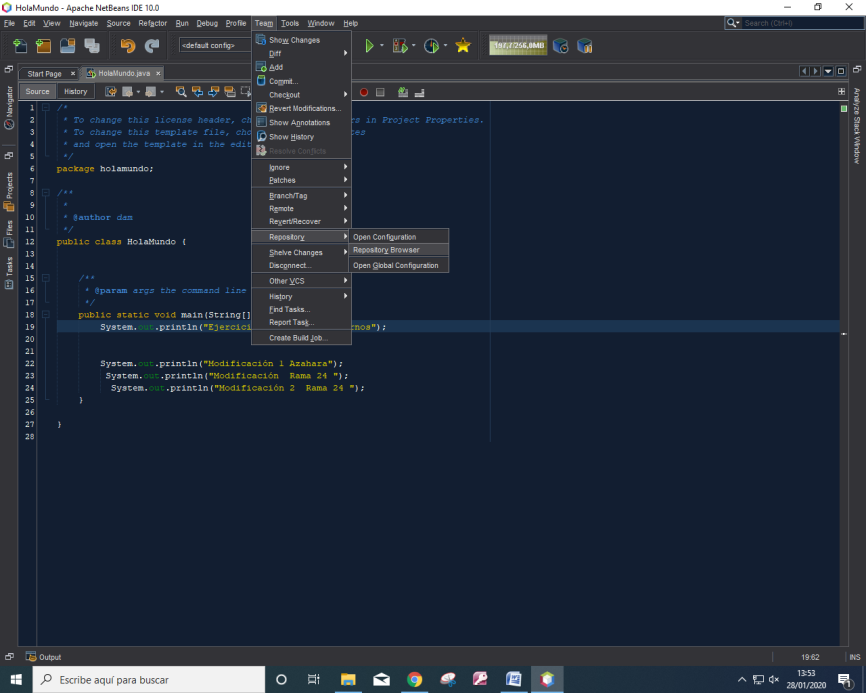
Hace dos cambios en el archivo HolaMundo.java. Después de estar conforme con el trabajo que ha hecho, decide fusionar la rama rama24 con la rama master en local.



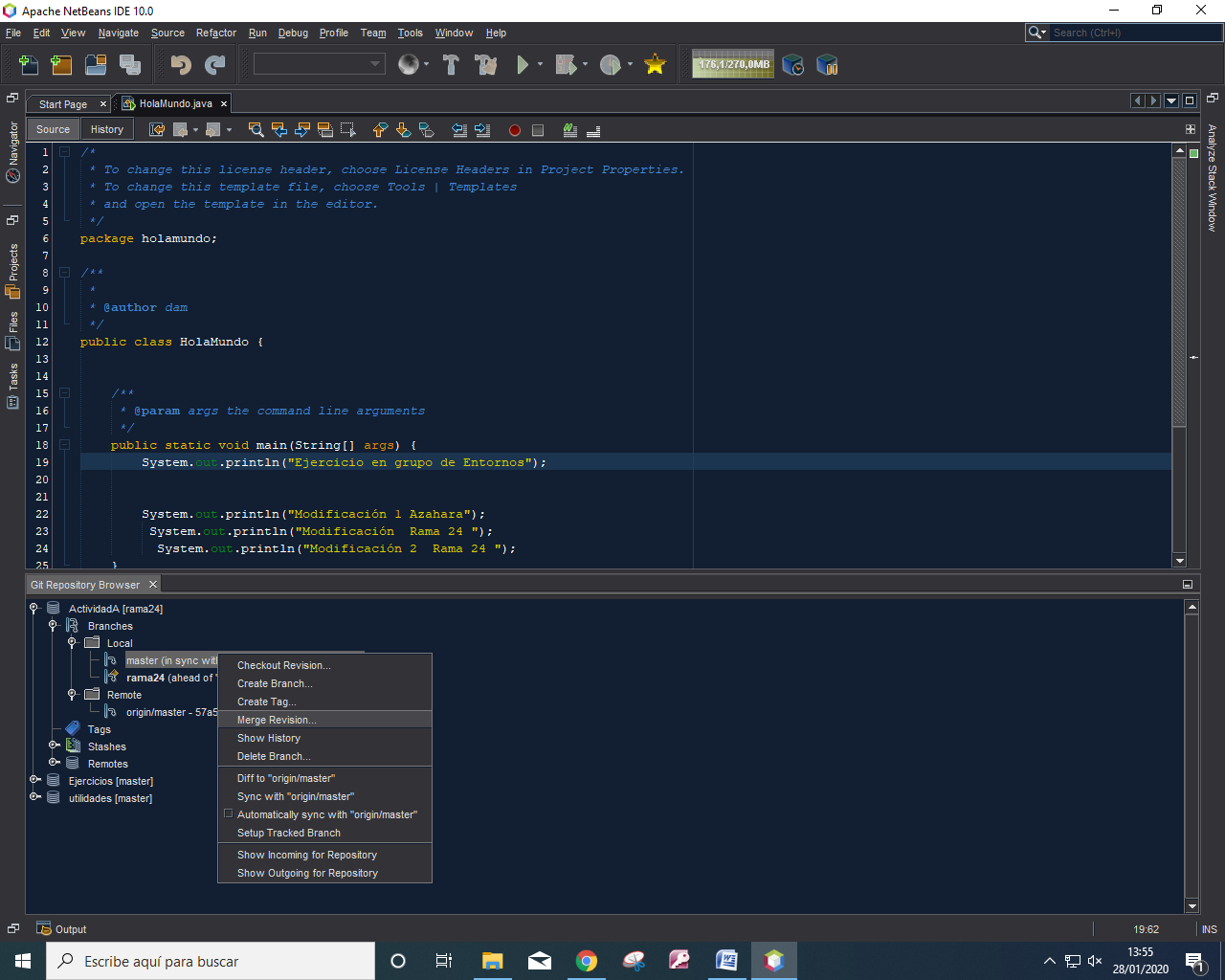
Siempre se realiza un Commit:







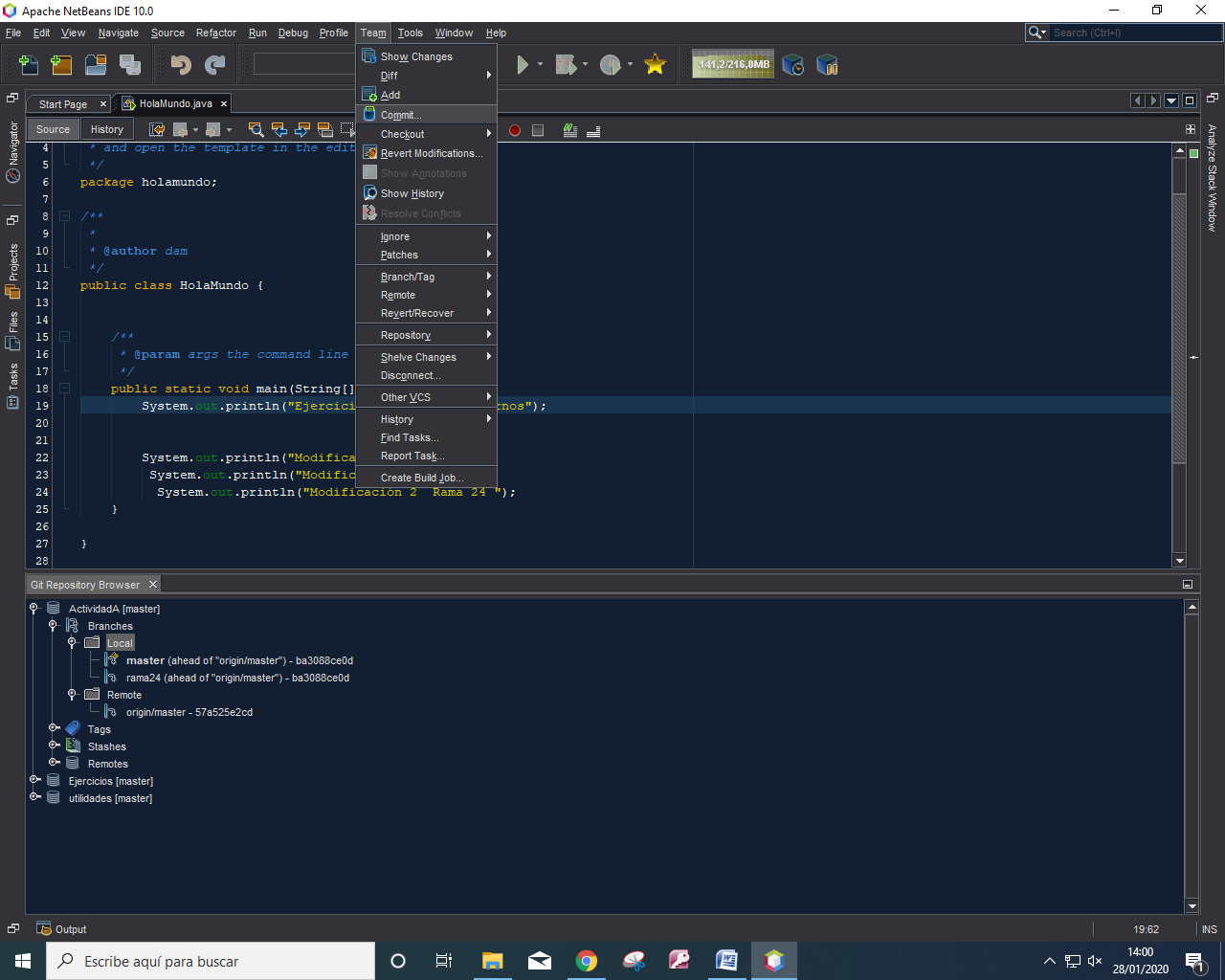
Team, Repositorio, Repositorio Browser para ver las ramas y ver sobre cual se está trabajando.

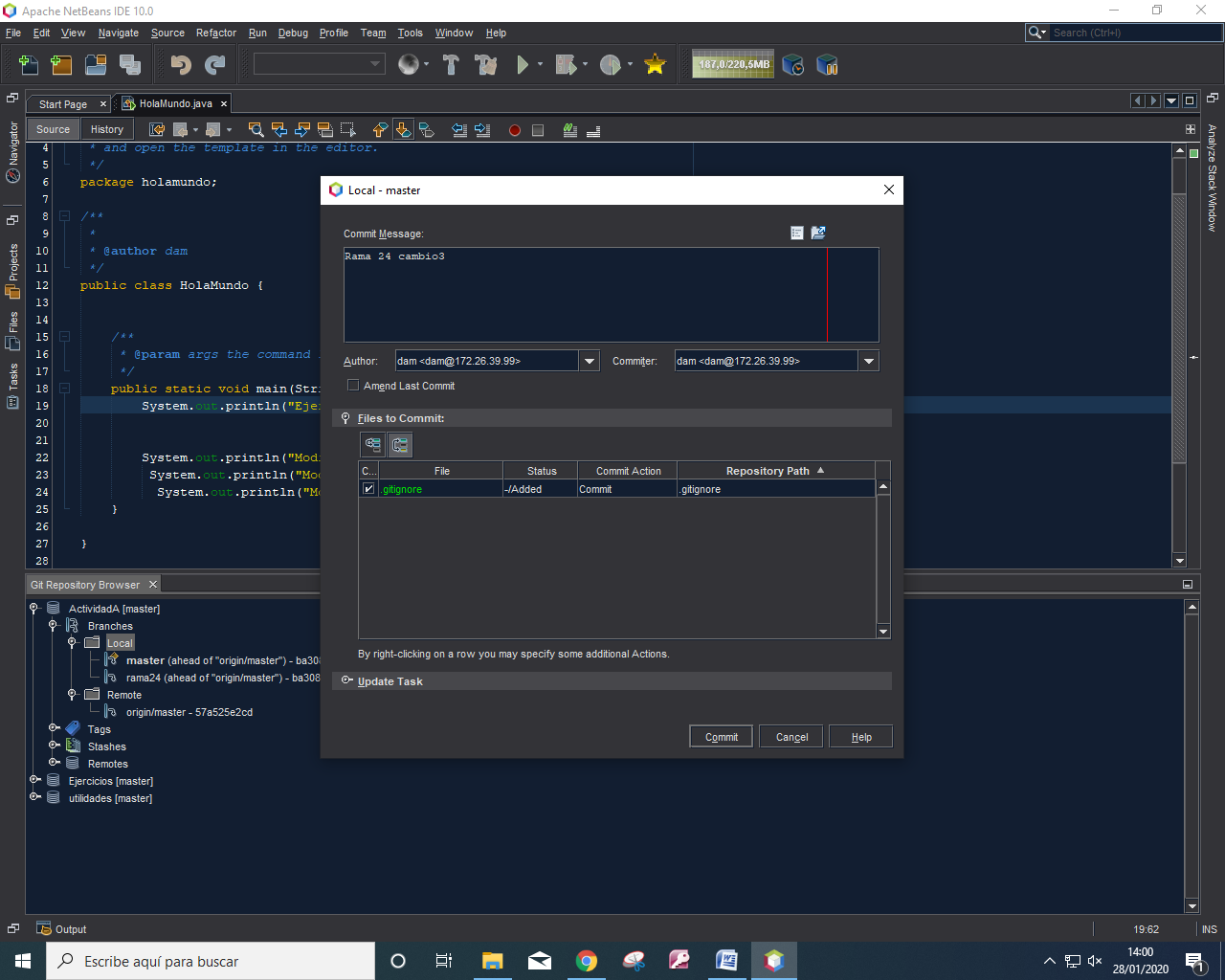


Estando en el master , seleccionamos la rama que queremos fusionar con el botón derecho del raton y selecionamos Merge Revision para fusionarlas.

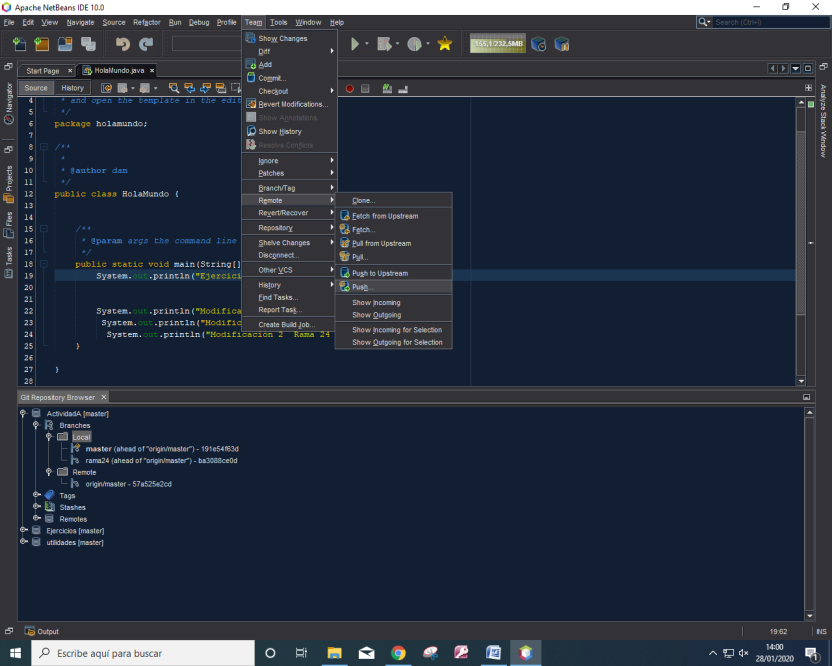
• Ahora sube los cambios al repositorio remoto. Indica los pasos que debes realizar.

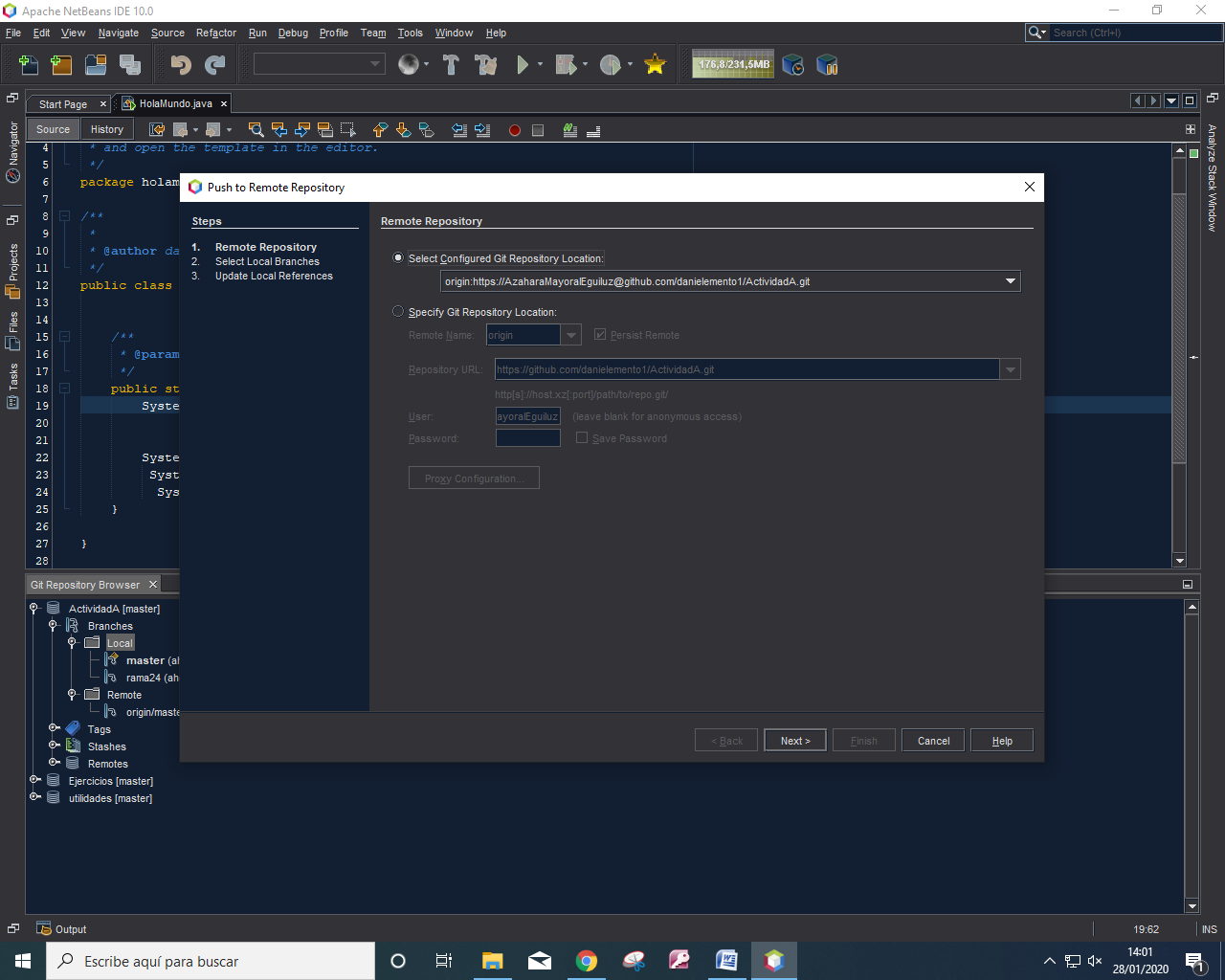
Para acabar se realiza un commit seguido de un push para actualizar el repositorio remoto con las ramas fusionadas.

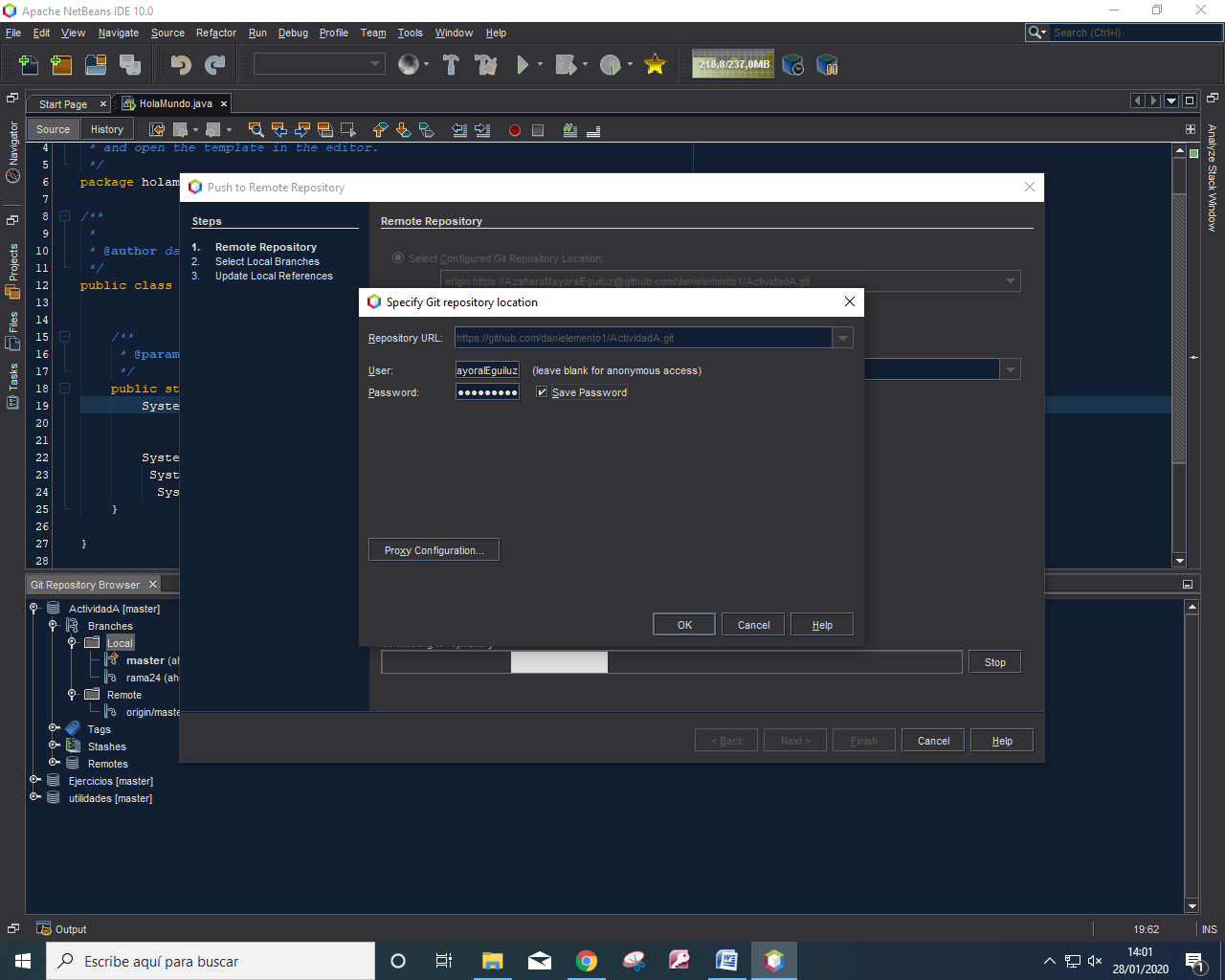


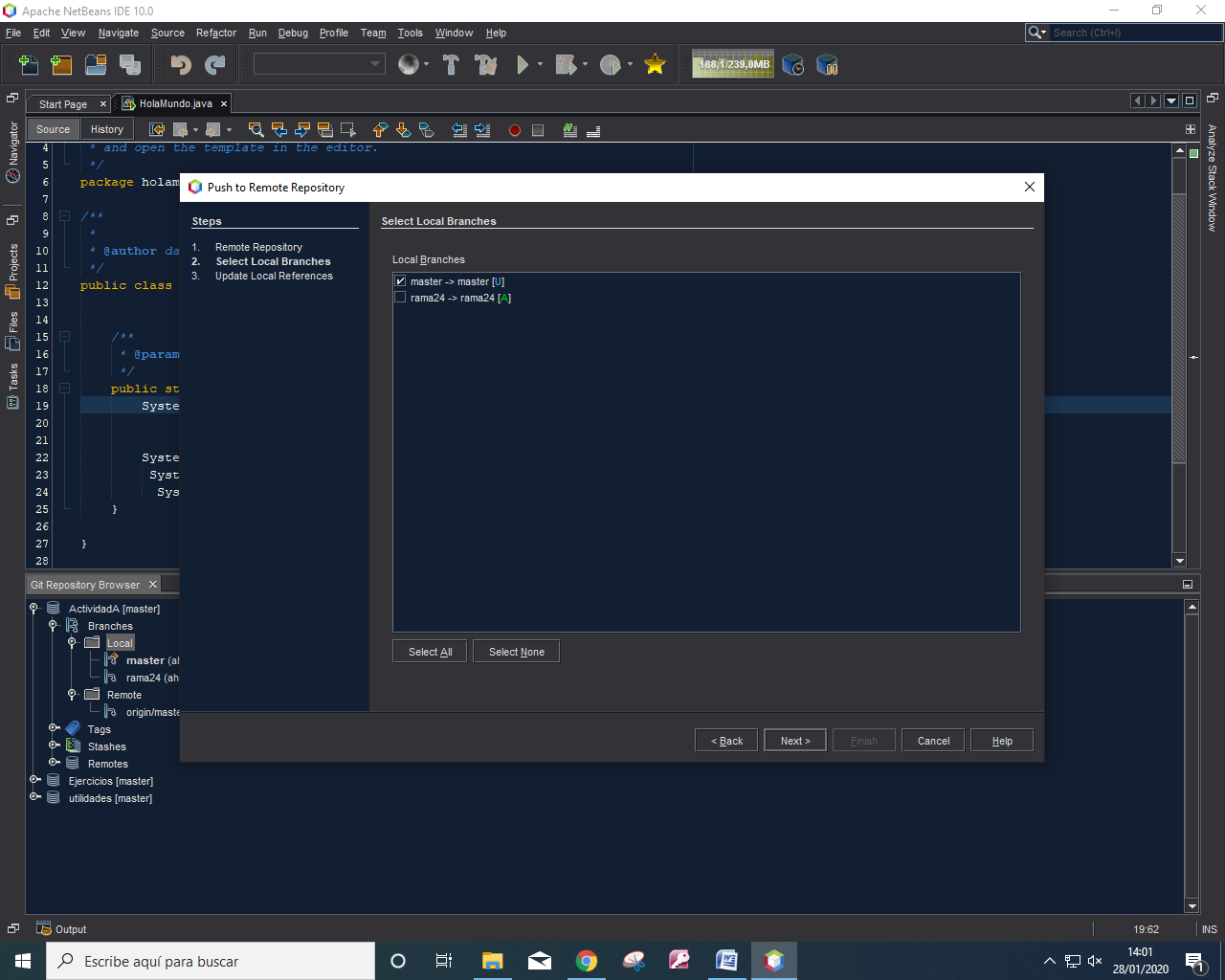


Push(Para subir los datos a Git):









Azahara solo sube la master para que vean el trabajo final y no los cambios uno a uno que ha ido añadiendo. Así en un trabajo en equipo les muestras lo definitivo y lo que sirve no llenas el Git con datos que no van a servir.

