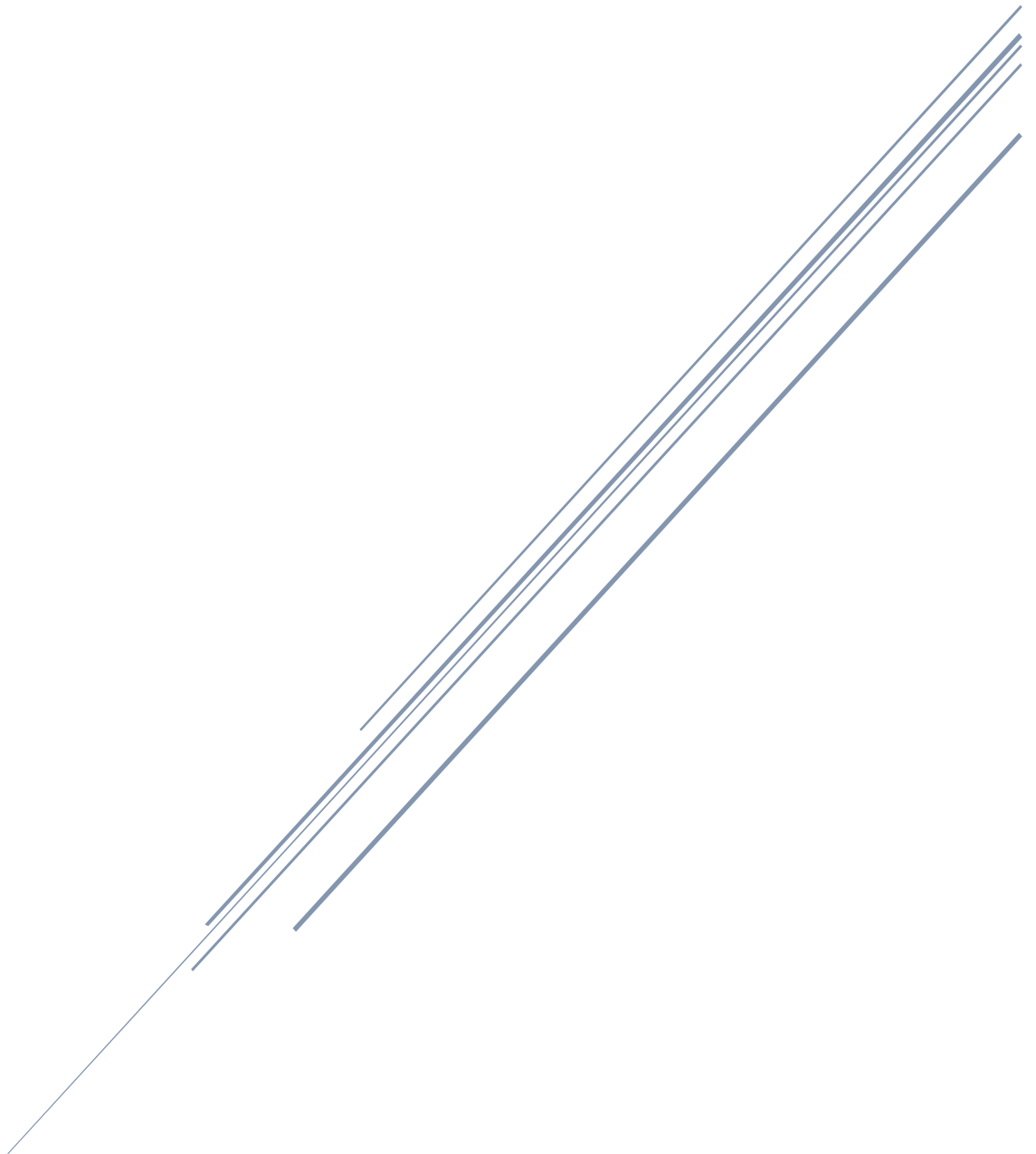


ED T4 Práctica II (Por parejas): Caso Práctico Git en NetBeans

Eduardo Martín-Sonseca y Mario Ortuñez



Gregorio Fernández
Desarrollo de Aplicaciones Multiplataforma

Contenido

1. Roles	2
2. Desarrollando ejercicio	2

1. Roles

Desarrollador 1- Eduardo Martín-Sonseca

Desarrollador 2- Mario Ortuñez

Ambos

2. Desarrollando ejercicio

Lo primero que hacen ambos es clonar el repositorio remoto cada uno en su equipo. Por defecto, los repositorios tanto remotos como locales sólo tienen una rama llamada main. De momento con esta rama tienen suficiente. Al realizar la clonación tendrán una copia en local de los archivos del proyecto conjunto. Dicho proyecto se copiará en la rama main del repositorio local. Durante el proceso de clonación el programa pregunta dónde se quiere ubicar el repositorio local, entre otras cosas. Una vez terminado el proceso NetBeans pregunta si se quiere crear un proyecto.

Desarrollador 1- Eduardo Martín-Sonseca

Para clonar el repositorio, nos dirigiremos a **Apache NetBeans 14**, dirigiéndonos a GitHub, y copiando el URL desde GitHub

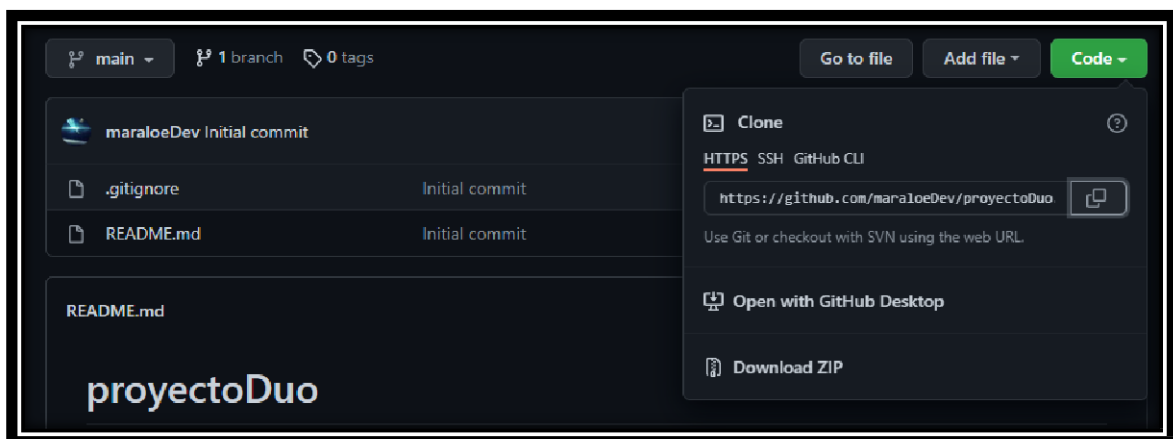


Ilustración 1

Una vez copiado, nos dirigiremos a **Team > Git > Clone...**

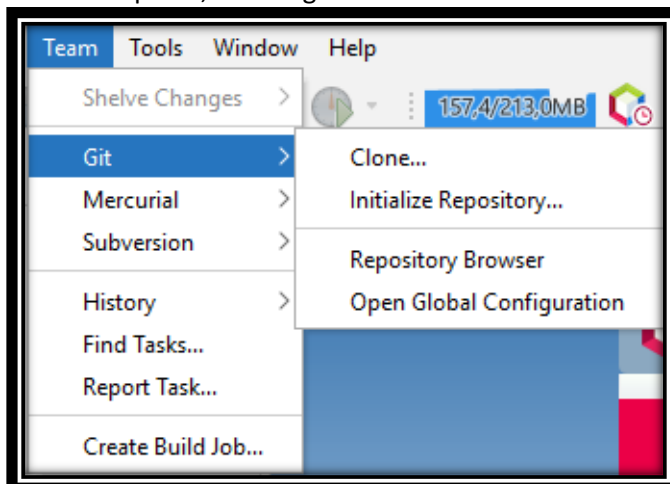


Ilustración 2

Una vez clicado, nos dirigiremos a la pestaña **Repository URL** en el que copiaremos la URL del repositorio que copiamos anteriormente, introduciremos nuestro usuario y contraseña de GitHub y señalaremos el lugar donde guardar los proyectos, una vez hecho eso, clicaremos en **next**

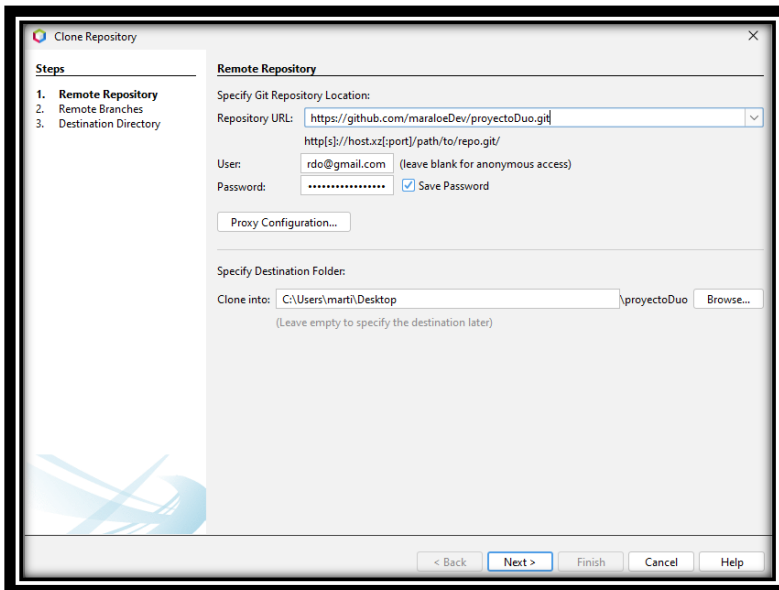


Ilustración 3

Una vez clicado en next, seleccionaremos la rama y el directorio donde se guardara

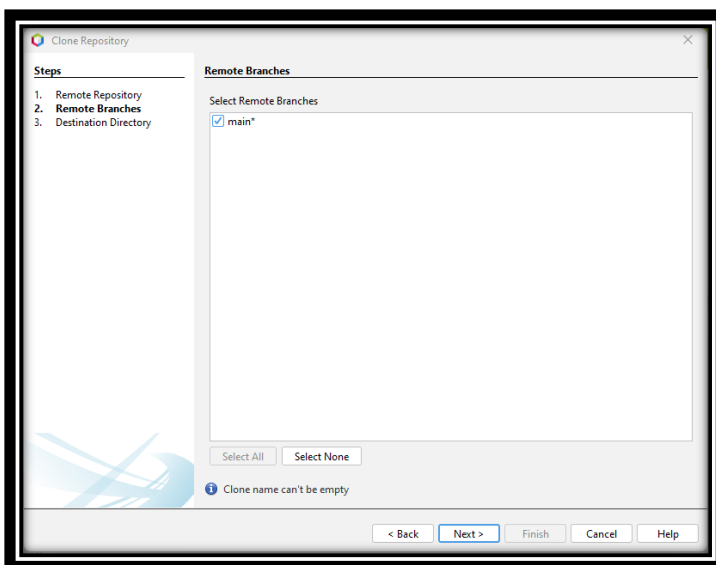


Ilustración 4

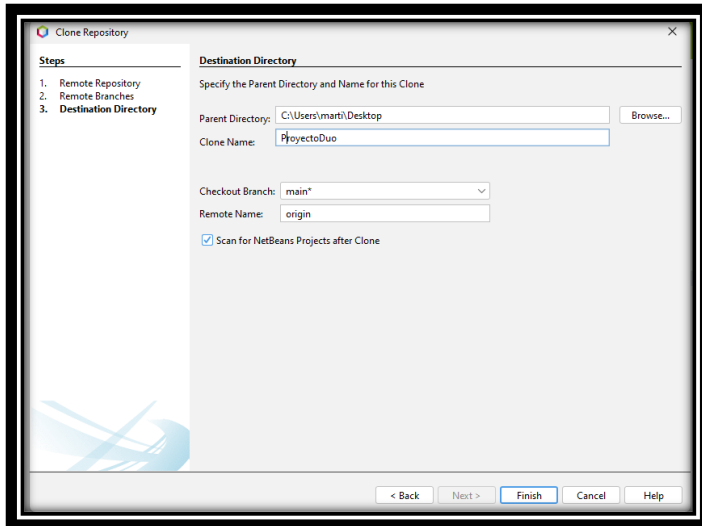


Ilustración 5

Una vez clonado, nos dirigiremos a crear un paquete llamado **HolaDAM**, y clicaremos en **Finish**

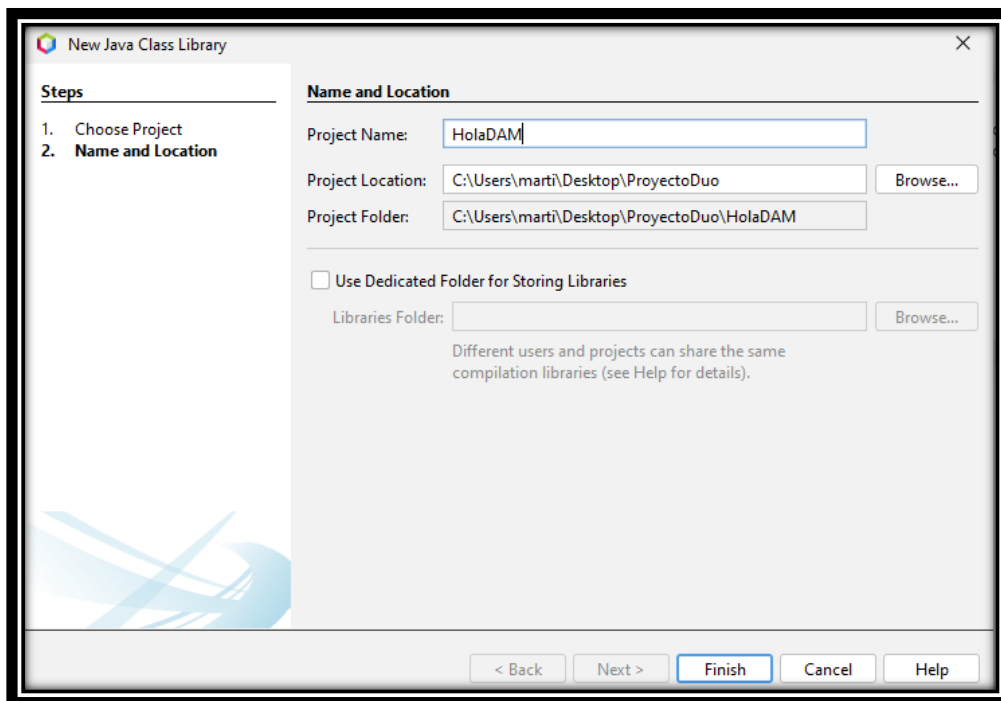
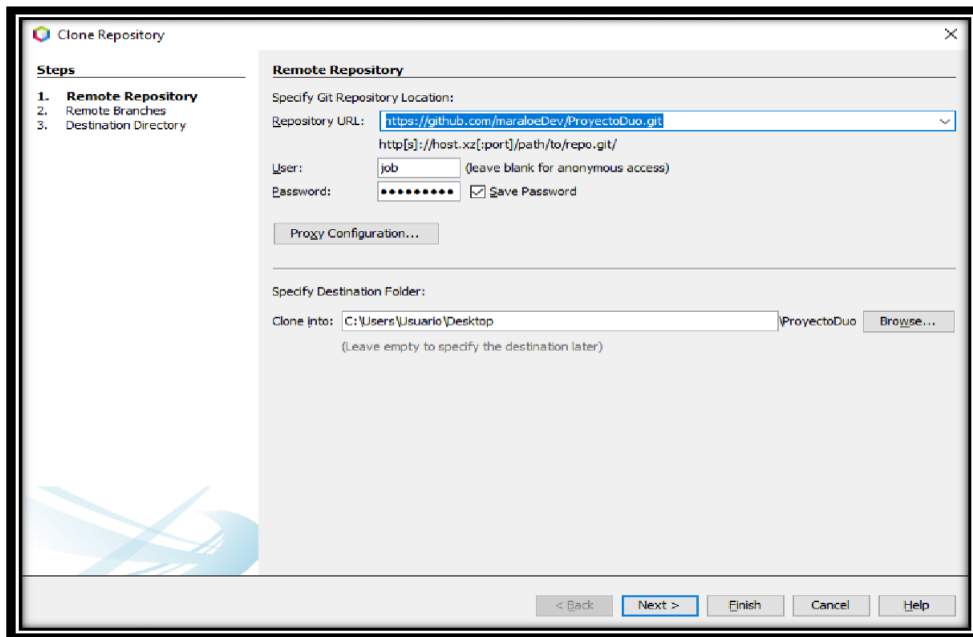


Ilustración 6

Desarrollador 2- Mario Ortuñez+

Ilustración 6



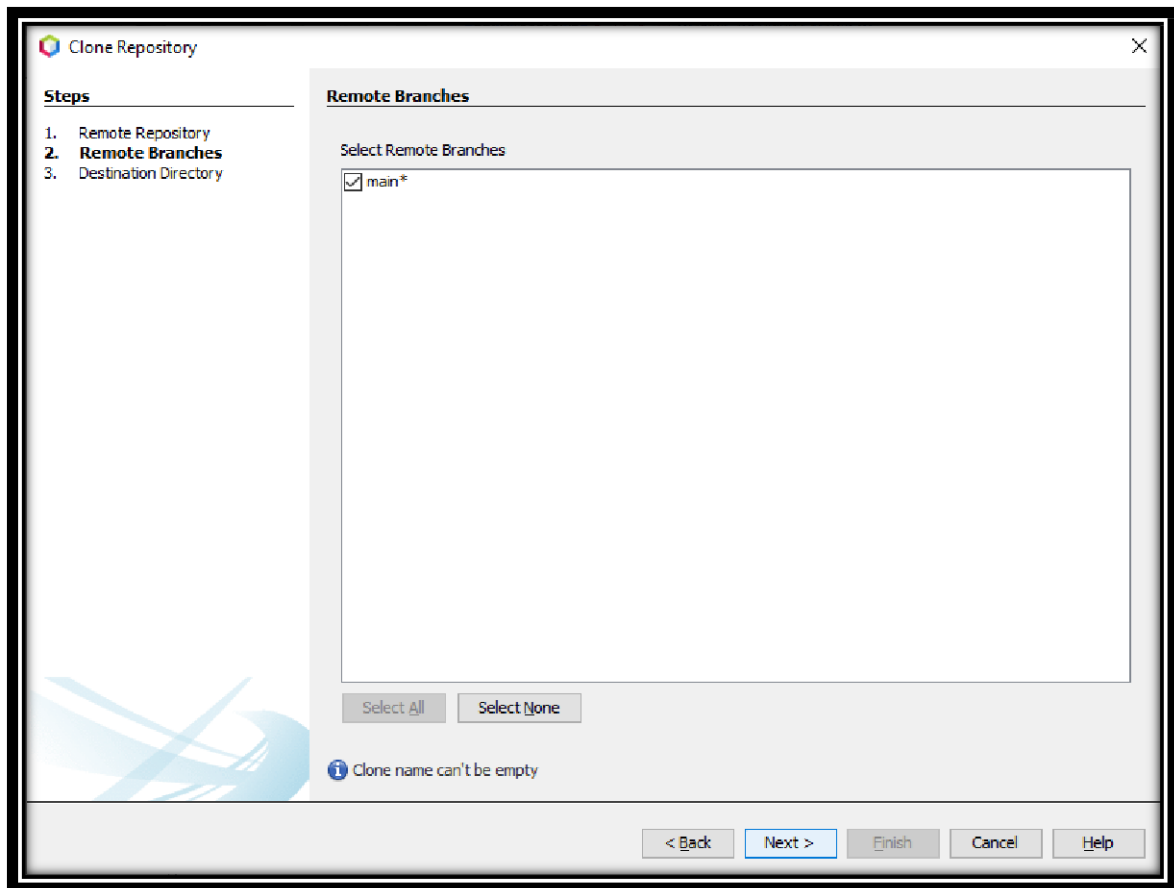


Ilustración 7

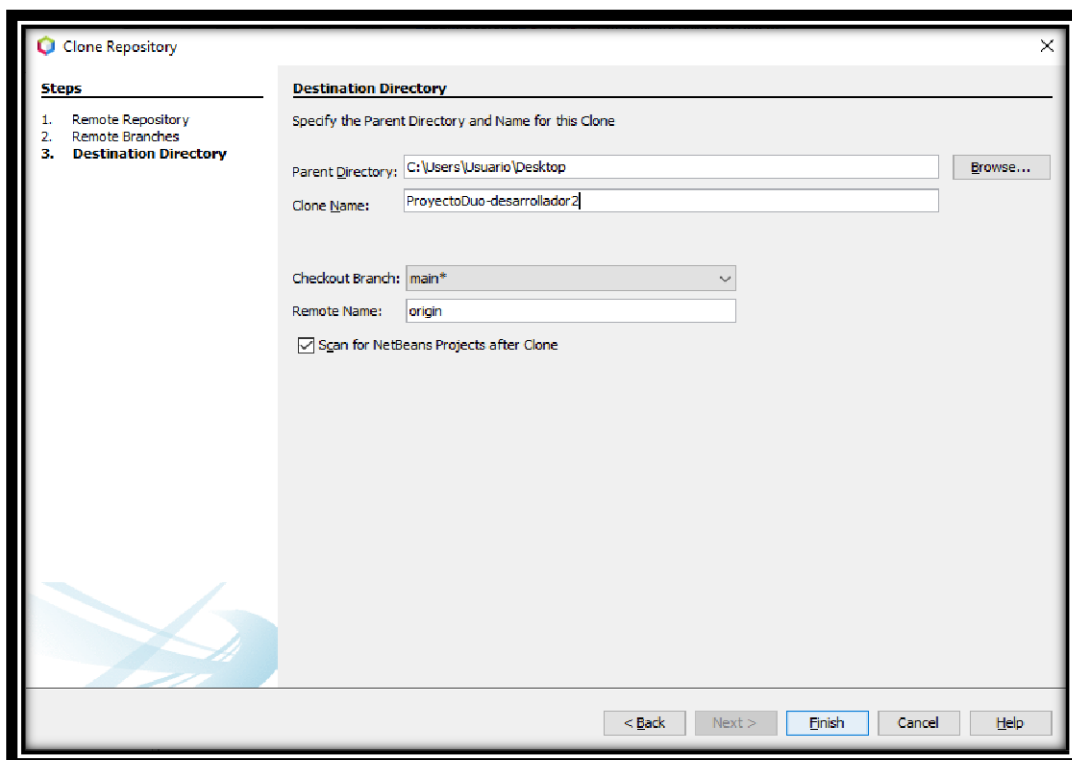


Ilustración 8

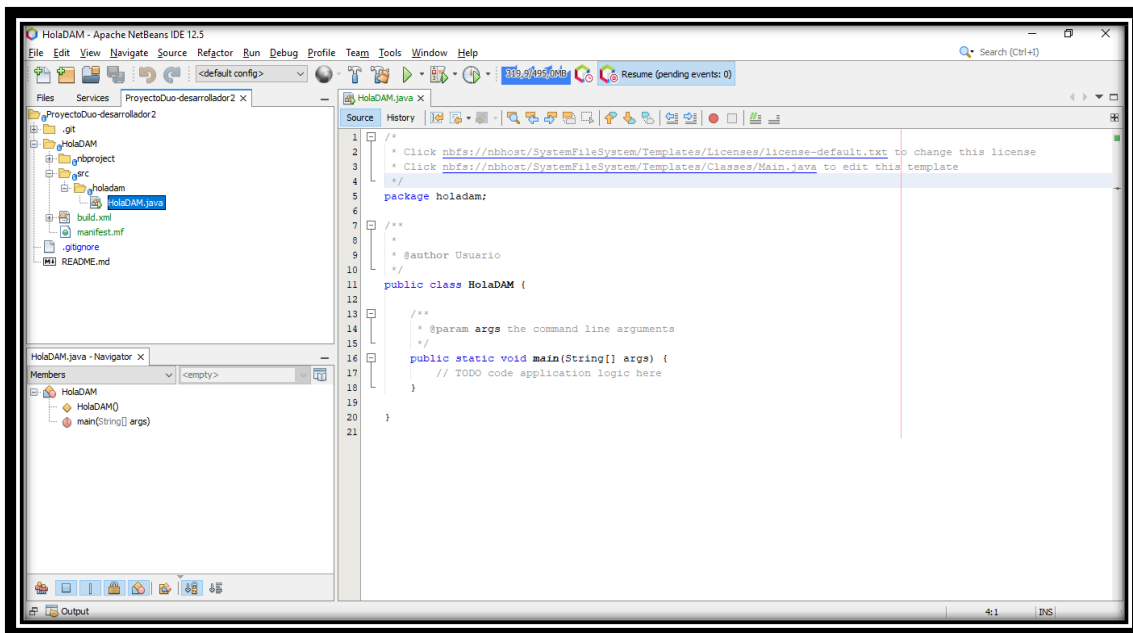


Ilustración 9

Uno de los programadores crea un proyecto llamado HolaDAM en el repositorio local. Se crea un archivo HolaDAM.java, el cual editará y subirá al repositorio remoto.

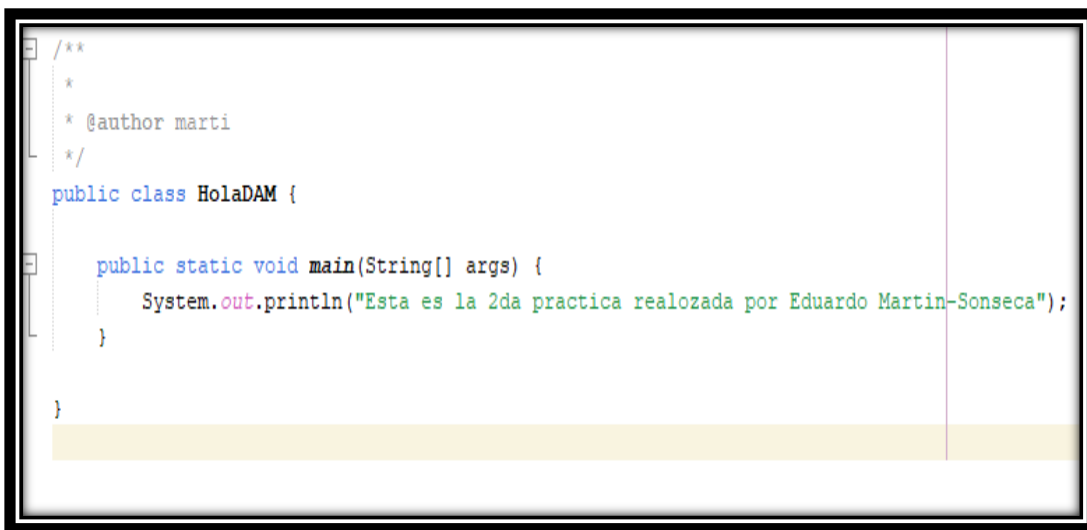


Ilustración 10

Una vez creado el documento, lo subiremos al repositorio remoto, haciendo clic derecho en el archivo creado y seleccionaremos **Team > Commit**, estableceremos una descripción, y clicaremos en **Commit**

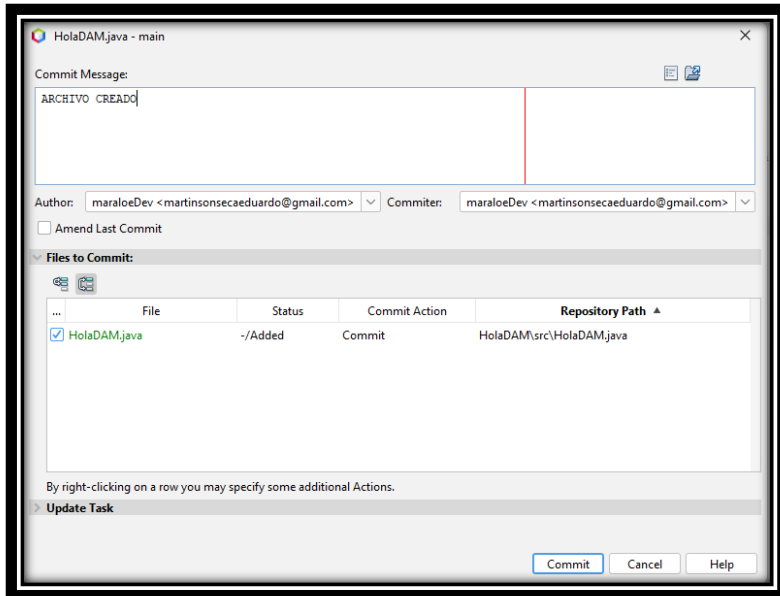


Ilustración 11

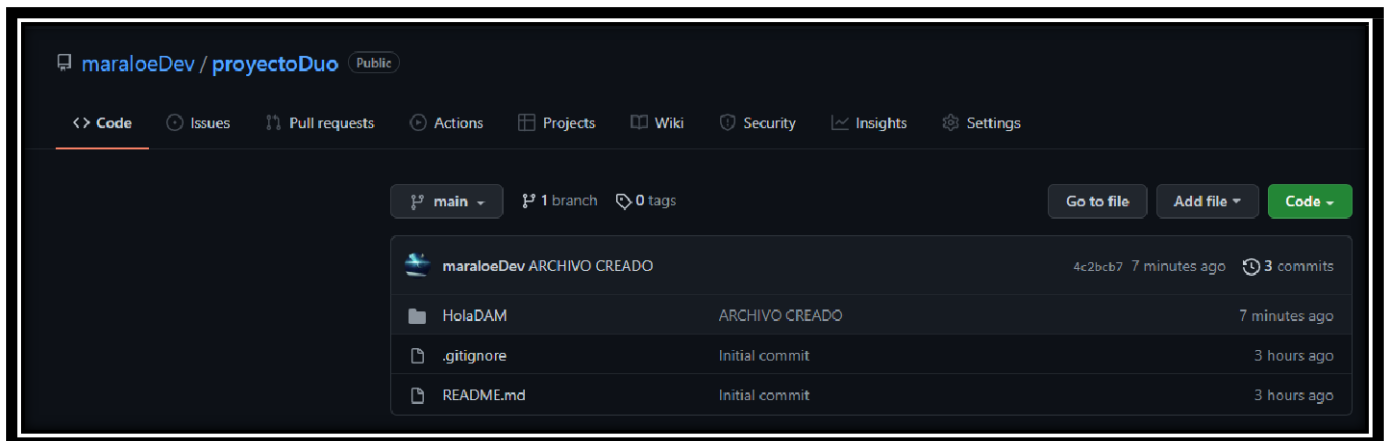


Ilustración 12

Una vez subido, confirmaremos que se ha subido correctamente, dirigiéndonos a **GitHub**

Ahora el otro desarrollador lo descarga del repositorio remoto, bajándose las últimas versiones de los archivos del proyecto. Así lo que está haciendo es bajarse las últimas versiones de los archivos del proyecto

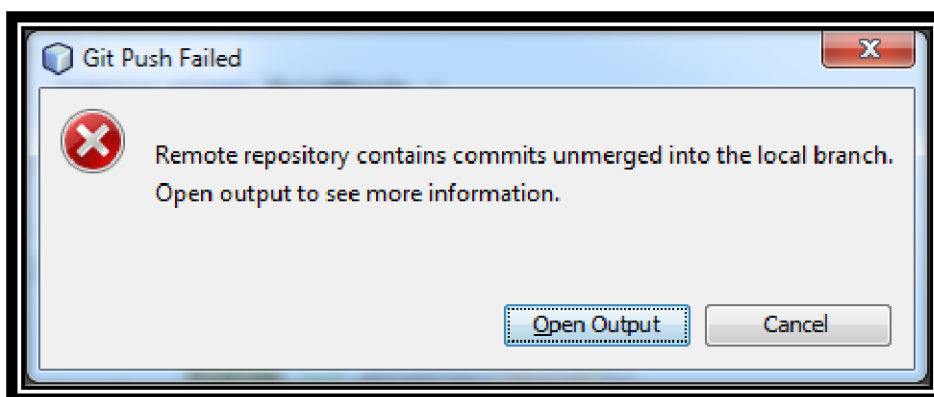


Ilustración 15

El primer desarrollador modifica el archivo HolaDAM.java de nuevo y lo vuelve a subir al repositorio remoto. No hay problemas al hacer la subida, ya que la versión de HolaDAM.java que estaba en el repositorio remoto debe de ser consistente con la que tenía el programador en local

Modificaremos el archivo, para cambiar el commit siguiendo los pasos anteriores

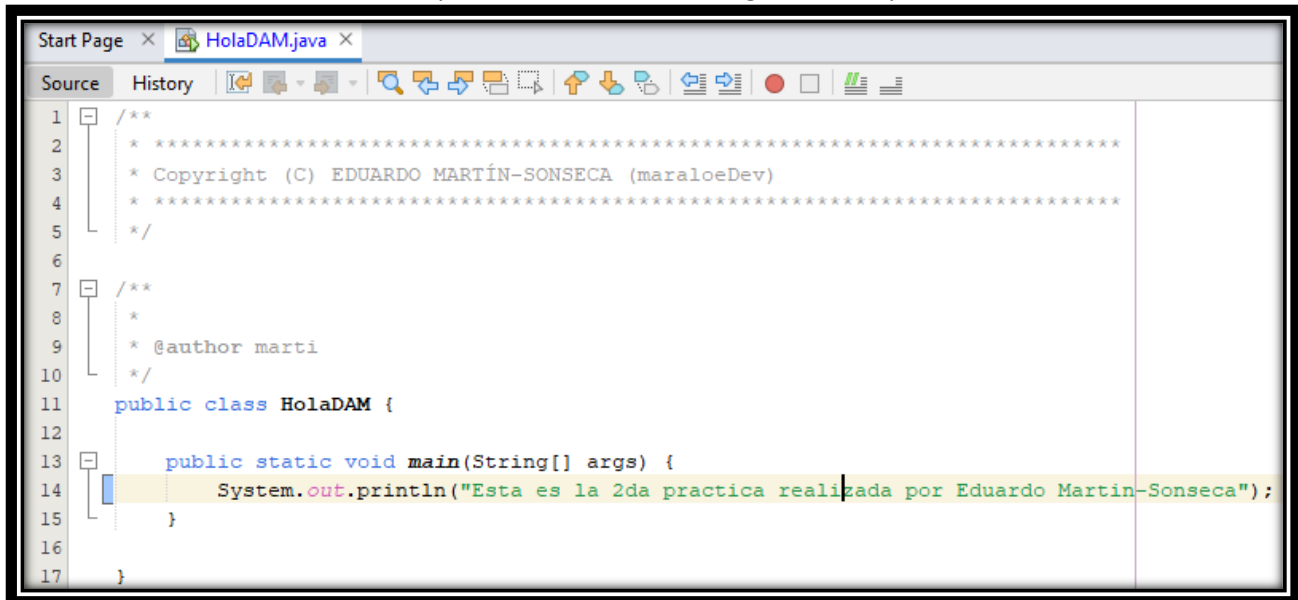


Ilustración 16

Comprobaremos que se ha hecho el commit dirigiéndonos a GitHub

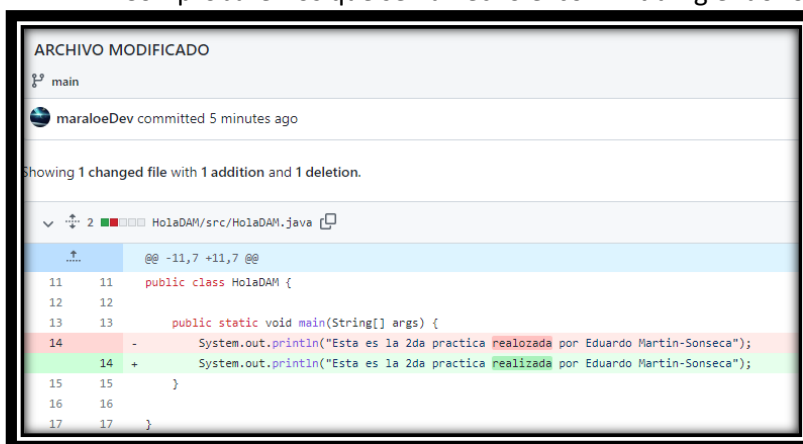


Ilustración 17

Mientras tanto, el segundo desarrollador modifica el archivo HolaDAM.java que tenía en su repositorio local. Al intentar subirlo obtendrá un mensaje de error. Esto es debido a que la versión de HolaDAM.java del repositorio remoto tiene modificaciones que no tenía la versión en local del desarrollador antes de que ella empezara a modificarlo. Es decir, los cambios que realizó el primer programador anteriormente no los tiene el segundo. (captura las pantallas).

Lo primero que tiene que hacer este desarrollador es bajarse una copia del repositorio remoto en local. Al hacerlo se creará una segunda rama en el repositorio local de él, llamada origin/main. Esto lo hace realizando la siguiente acción:

(*) Pista: Git → Remote → Fetch

A continuación, comprueba las inconsistencias entre los archivos locales que están en la rama main con los que están en la rama origin/main.

(*) Pista: Git → Branch/Tag → Merge revisión

Le da a Review para resolver los conflictos del mismo modo en que lo hacía por comandos. Una vez resueltos los conflictos, el desarrollador debe hacer add, commit y push para subir los archivos corregidos.

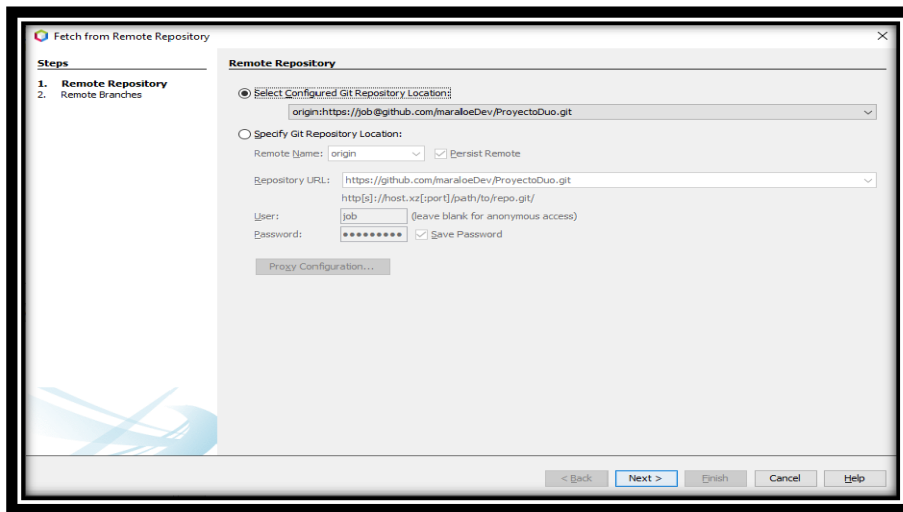


Ilustración 18

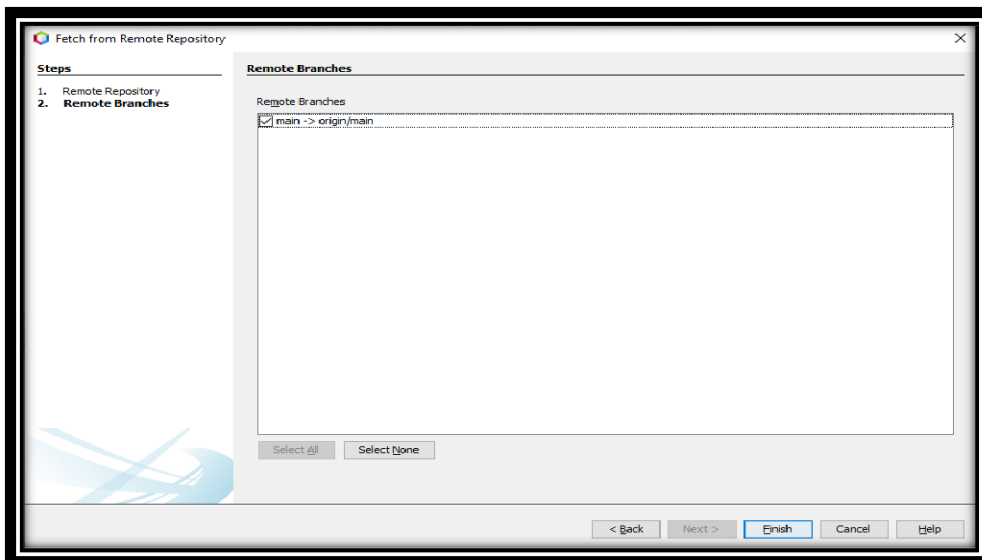


Ilustración 19

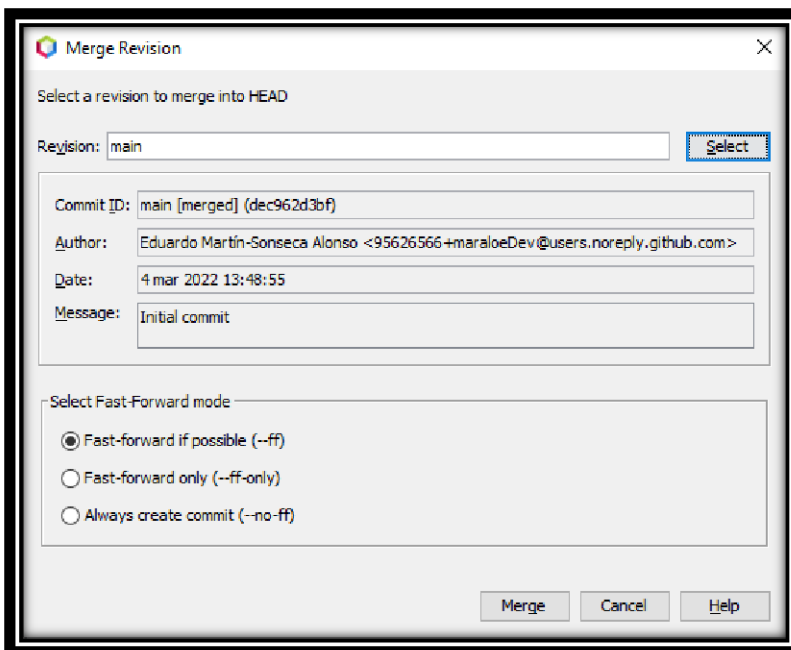


Ilustración 20

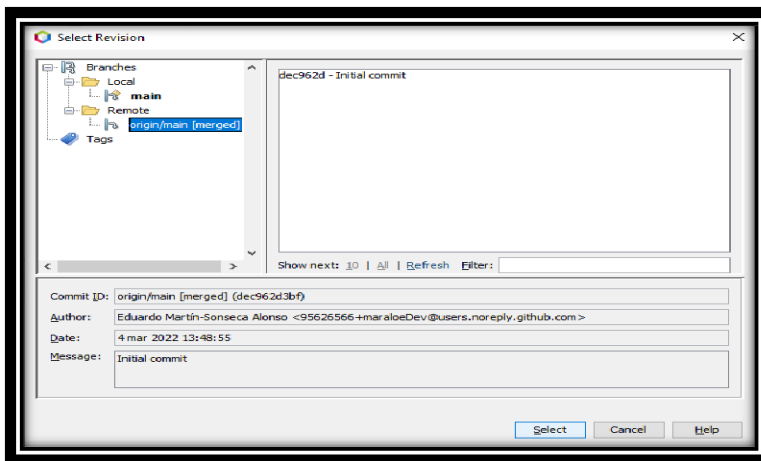


Ilustración 21

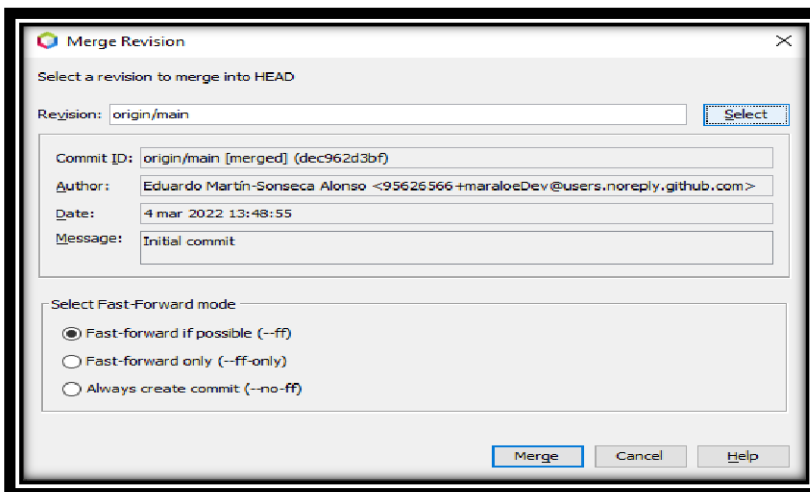


Ilustración 22

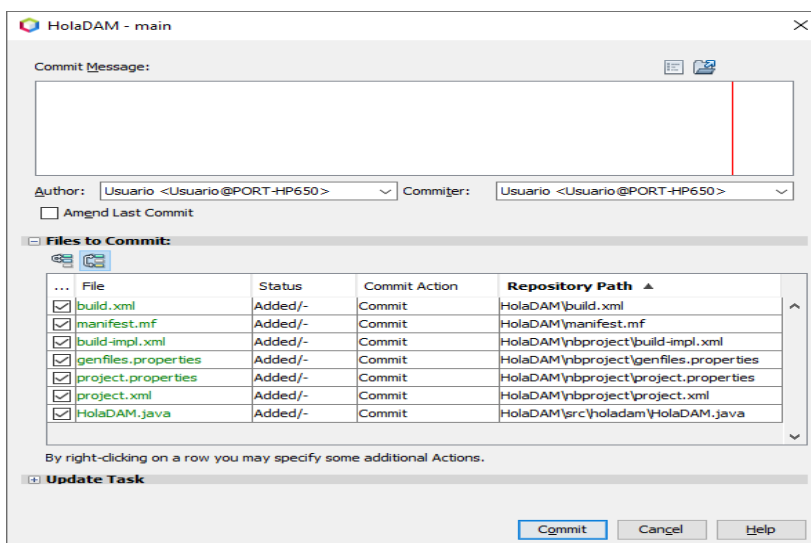


Ilustración 23

Mientras tanto, el primer desarrollador ha estado trabajando en una rama local aparte, llamada request23, creo la rama y la puse como activa. Ahora tiene dos ramas en su repositorio local; main y request23. Al crear la nueva rama, lo que ha hecho es crear una copia de la rama main local. Al activarla, ahora cualquier cambio que haga lo hará en dicha rama, quedándose la rama main en local sin cambios. Ahora el programador ha hecho dos cambios en el archivo HolaDAM.java (ha hecho dos adds y dos commits). Después de estar conforme con el trabajo que ha hecho, tiene que fusionar la rama request23 con la rama main en local. A continuación, fusionará ambas ramas para que la rama main tenga los cambios realizados en la rama request23

Para crear una nueva rama, nos dirigiremos a **Team > Branch > Create branch**

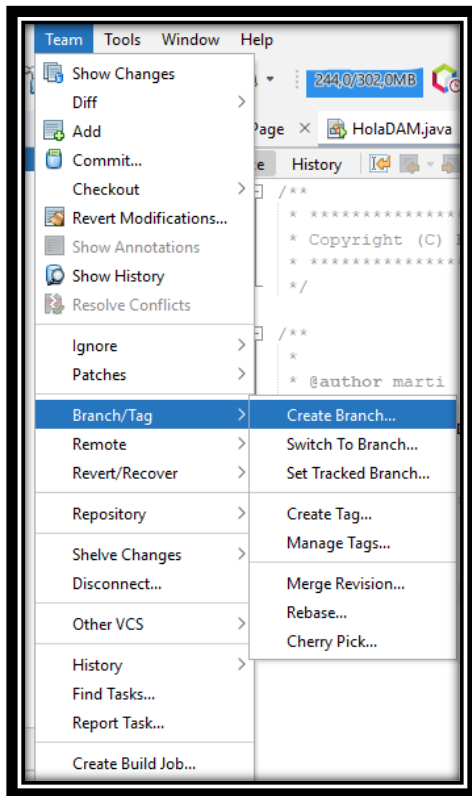


Ilustración 24

Una vez clicado, nos aparecerá un cuadro de diálogo, en el cual, denominaremos como se llamará la nueva rama

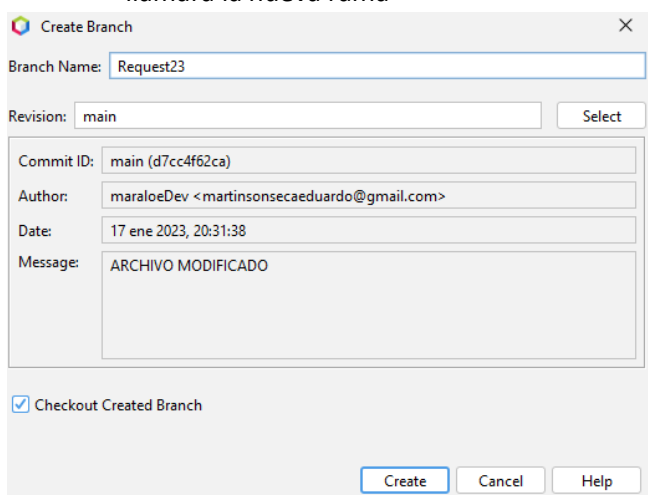


Ilustración 25

Una vez creada, modificaremos el archivo y lo commitearemos a la nueva rama (request23), confirmaremos los cambios dirigiéndonos a GitHub, para comprobar la rama creada y el archivo modificado en la rama

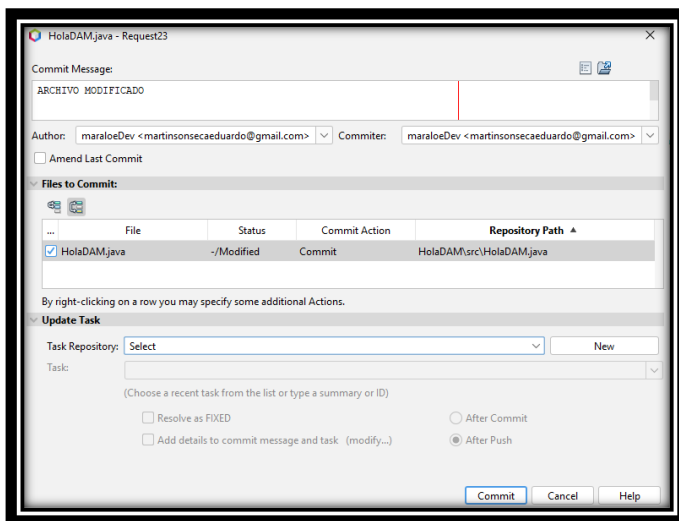


Ilustración 26

Una vez creada la rama, y la comprobación en GitHub, lo fusionamos, haciendo clic **Team > Branch tag > Switch to Branch** seleccionaremos la rama a fusionar (En este caso main con request23)

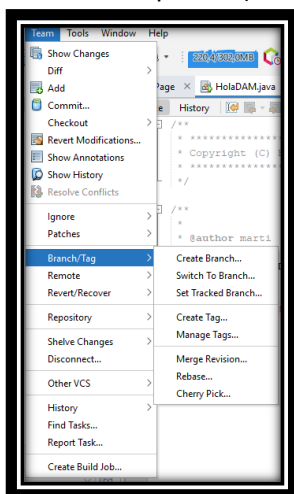


Ilustración 27

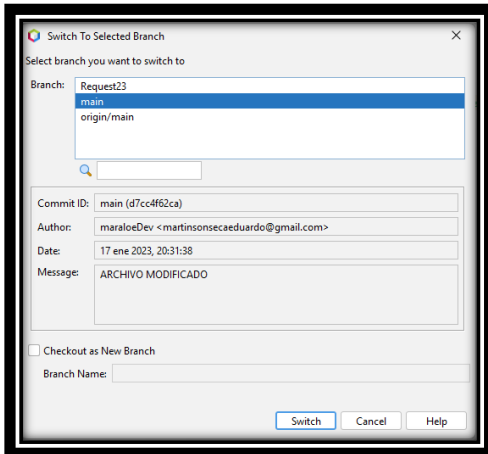


Ilustración 28

Una vez fusionado, lo compararemos con la otra rama, a ver si han surgido los cambios

Ahora tiene que subir los cambios al repositorio remoto. Lo primero tiene que hacer es un fetch con la rama main del repositorio remoto. De esta manera se crea la rama local origin/main con los archivos ubicados en la rama main del repositorio remoto. Ya solo le queda fusionar los cambios entre ambas ramas. Los cambios se fusionarán. Ya sólo quedaría hacer un push. Después de la ejecución del último comando, en el repositorio remoto, en la rama main, se encuentra el proyecto conjunto de los dos desarrolladores totalmente consistente y con el trabajo de ambos actualizado.

Para hacer una recogida de datos, haremos clic derecho en el archivo, y seleccionaremos la opción **Git > Fetch**

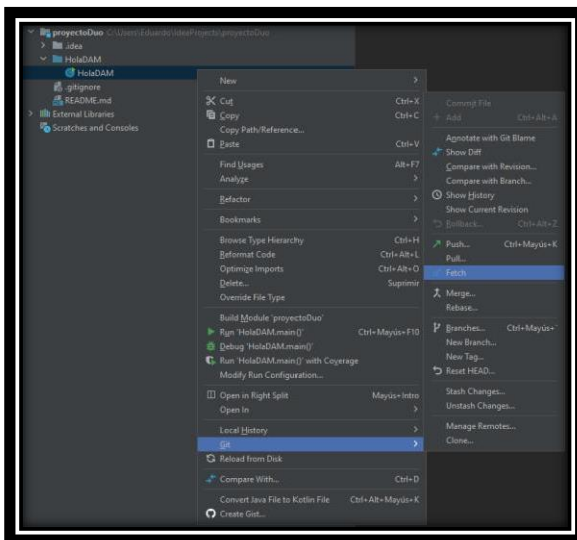


Ilustración 29

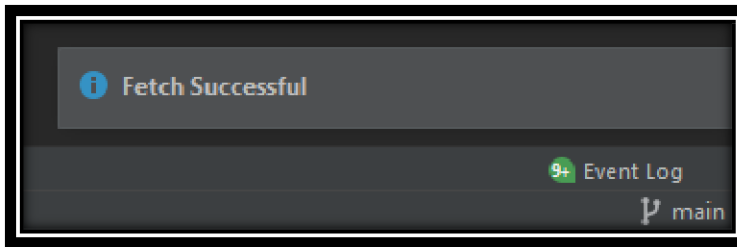


Ilustración 30

Para fusionar los cambios entre las 2 ramas

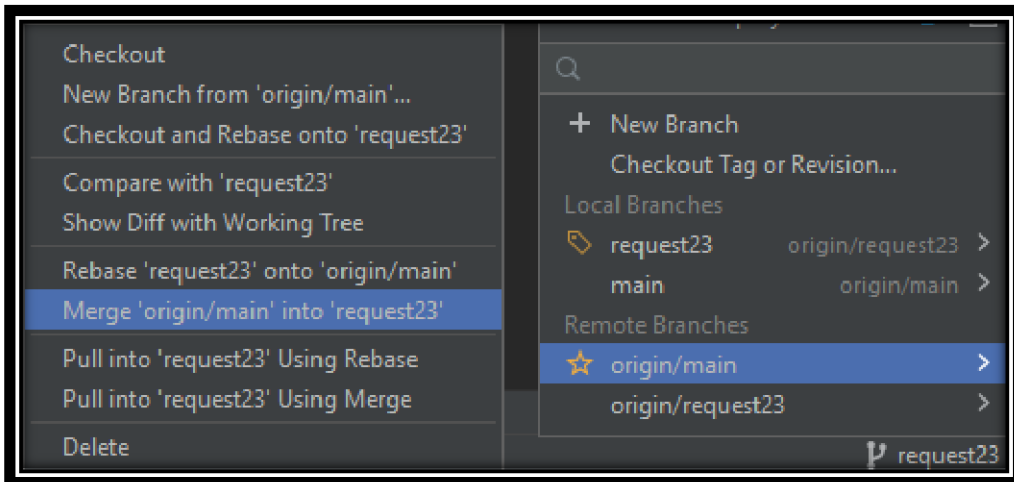


Ilustración 31