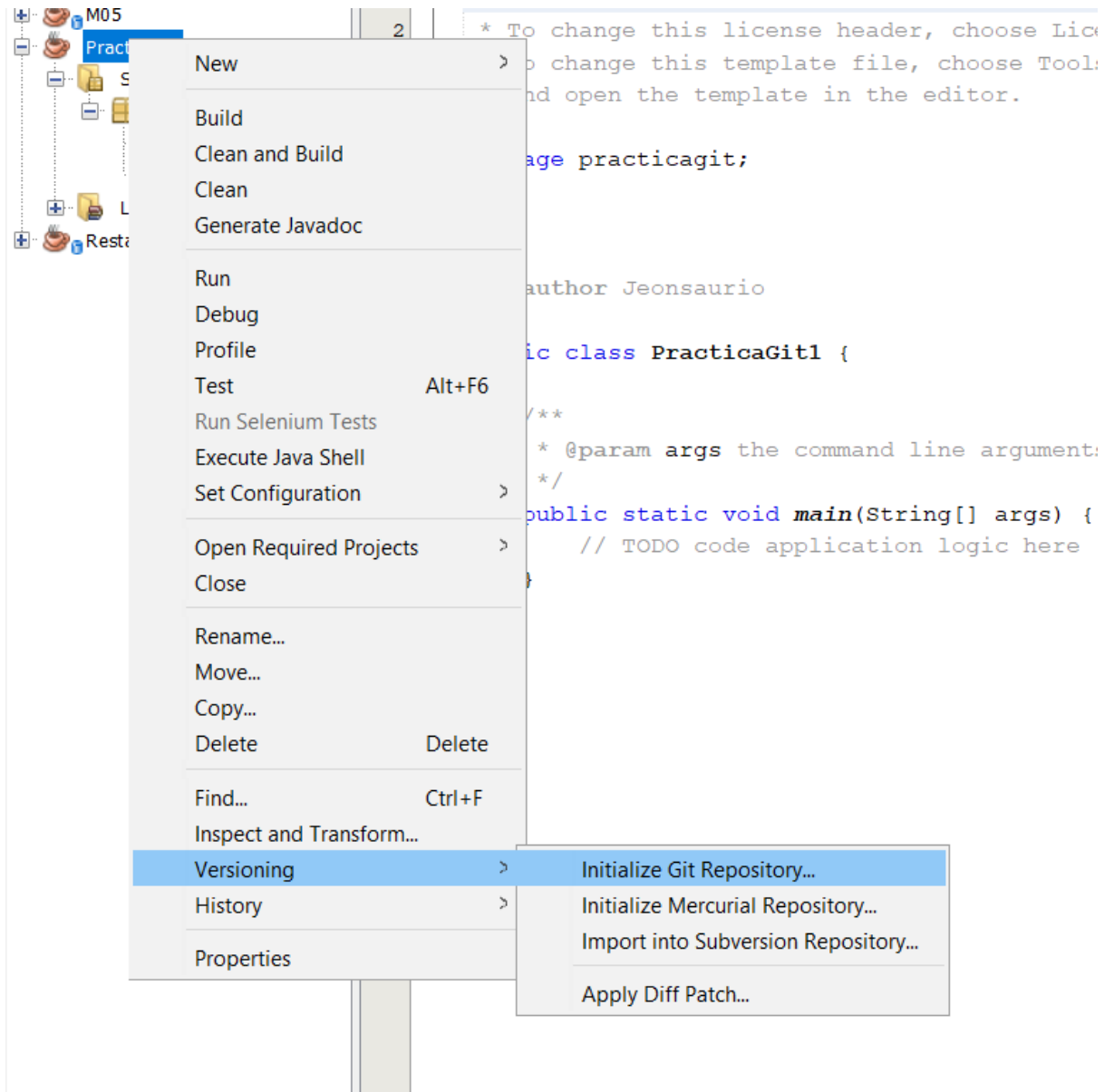


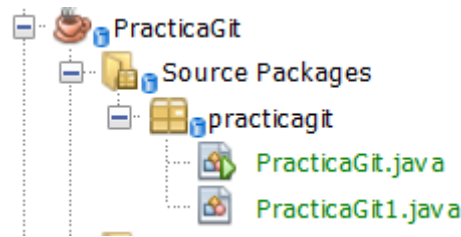
NETBEANS & GIT

Para poder utilizar el GIT del NetBeans, empezaremos con darle click derecho al proyecto en el que queramos trabajar, y seguidamente buscamos hasta la opción de Versioning e Inicializar el repositorio Git.

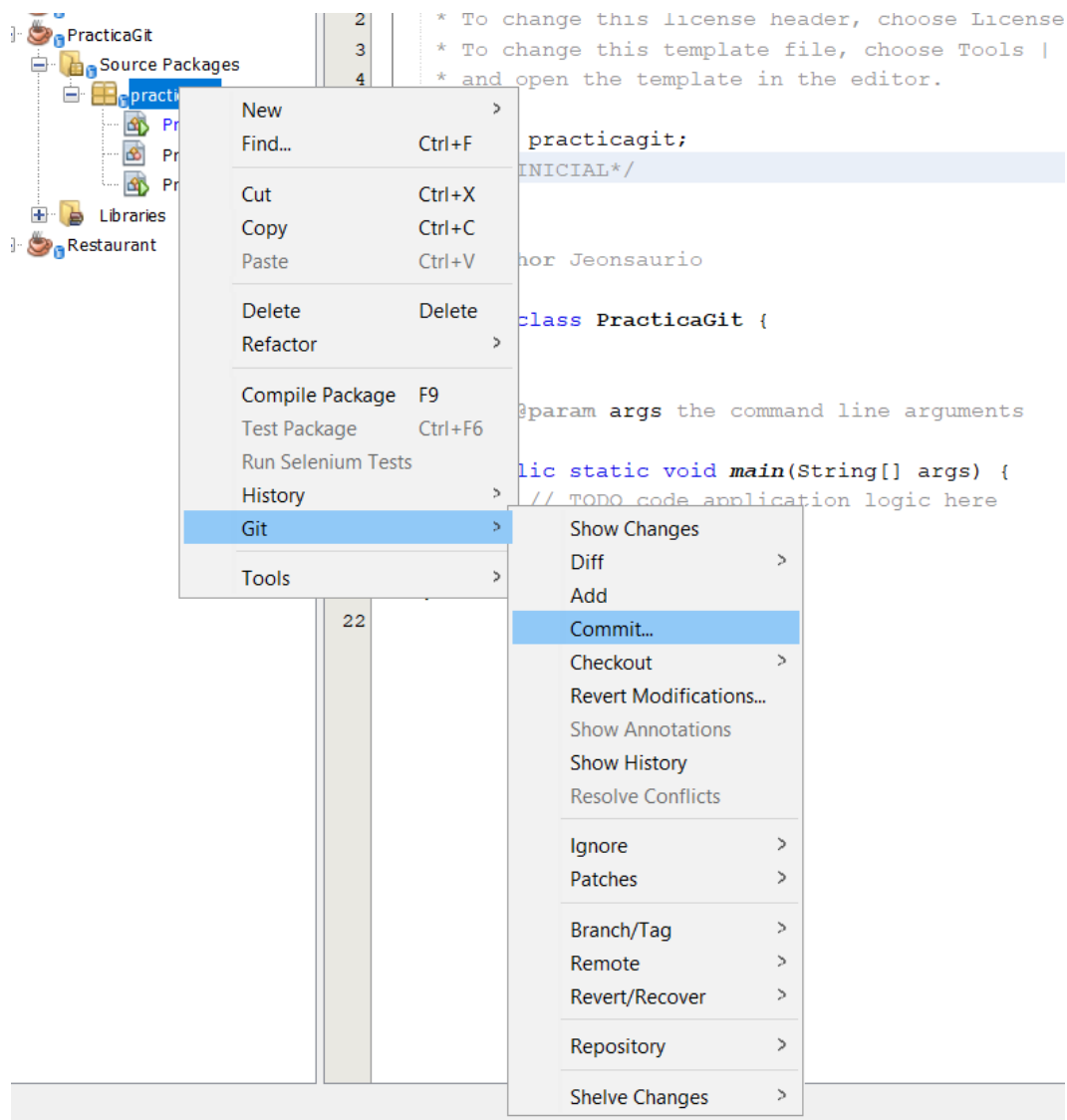


Ésto lo que hará es ir guardando todos los pasos que hemos realizado en los documentos dentro de los paquetes del proyecto.

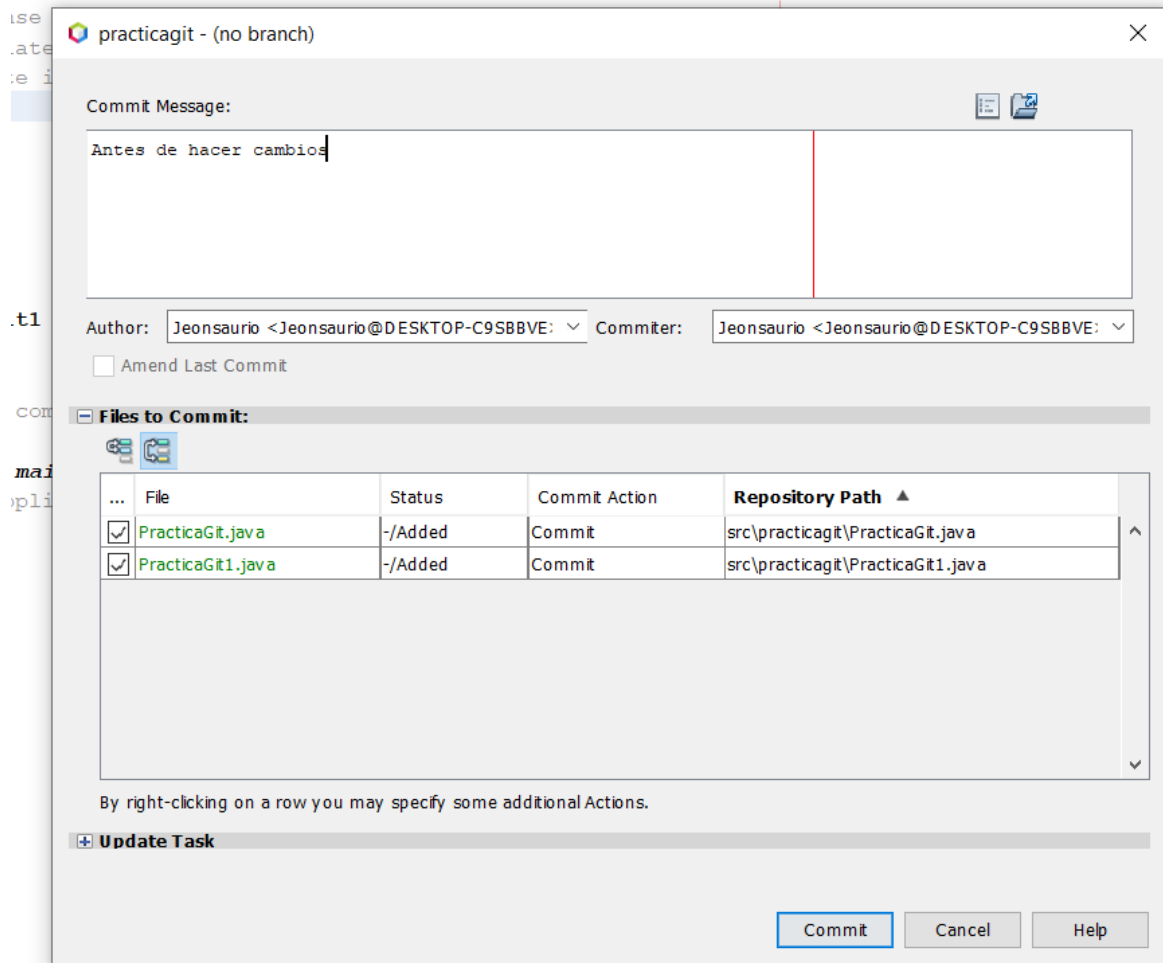
Vemos que los documentos dentro de los paquetes se han vuelto de color verde, y los paquetes tienen un pequeño icono azul, eso quiere decir que no están guardados dentro de lo que es el GIT.



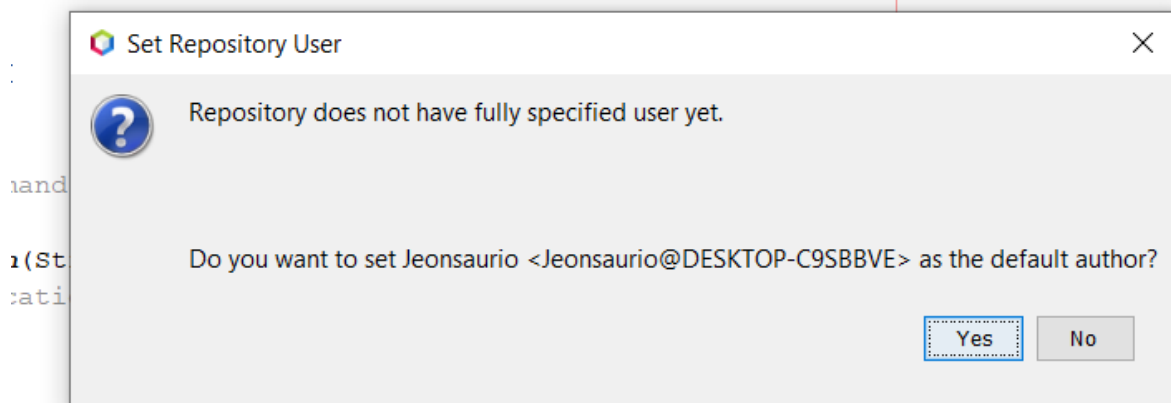
Lo que vamos a hacer ahora es guardar la información utilizando el "Commit". Volvemos a darle click derecho al paquete y vamos a la opción de Git y seleccionamos la pestaña de Commit.



El Commit puede hacer una copia de todos los ficheros o solamente uno.

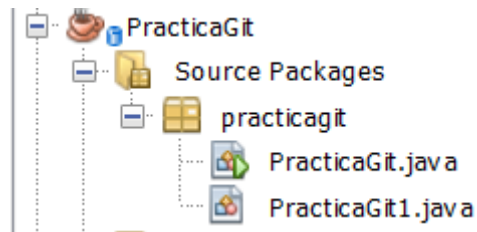


Es importante hacer un comentario antes de realizar el guardado para saber qué cambios has realizado.



Le damos a la opción de Yes.

Podemos ver que ya nuestro package no tiene ningún icono y nuestros archivos se han vuelto de color negro, eso quiere decir que el GIT ya tiene registrados los archivos al momento.



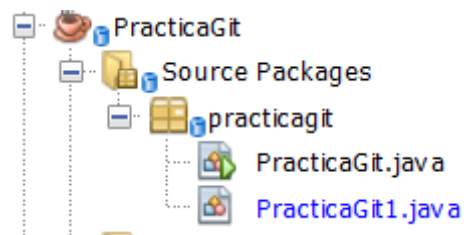
Sin embargo, si hacemos algún cambio en alguno de los archivos creados, éste se volverá de color azul, tanto si lo modificamos, cambiamos el nombre, etc...

```

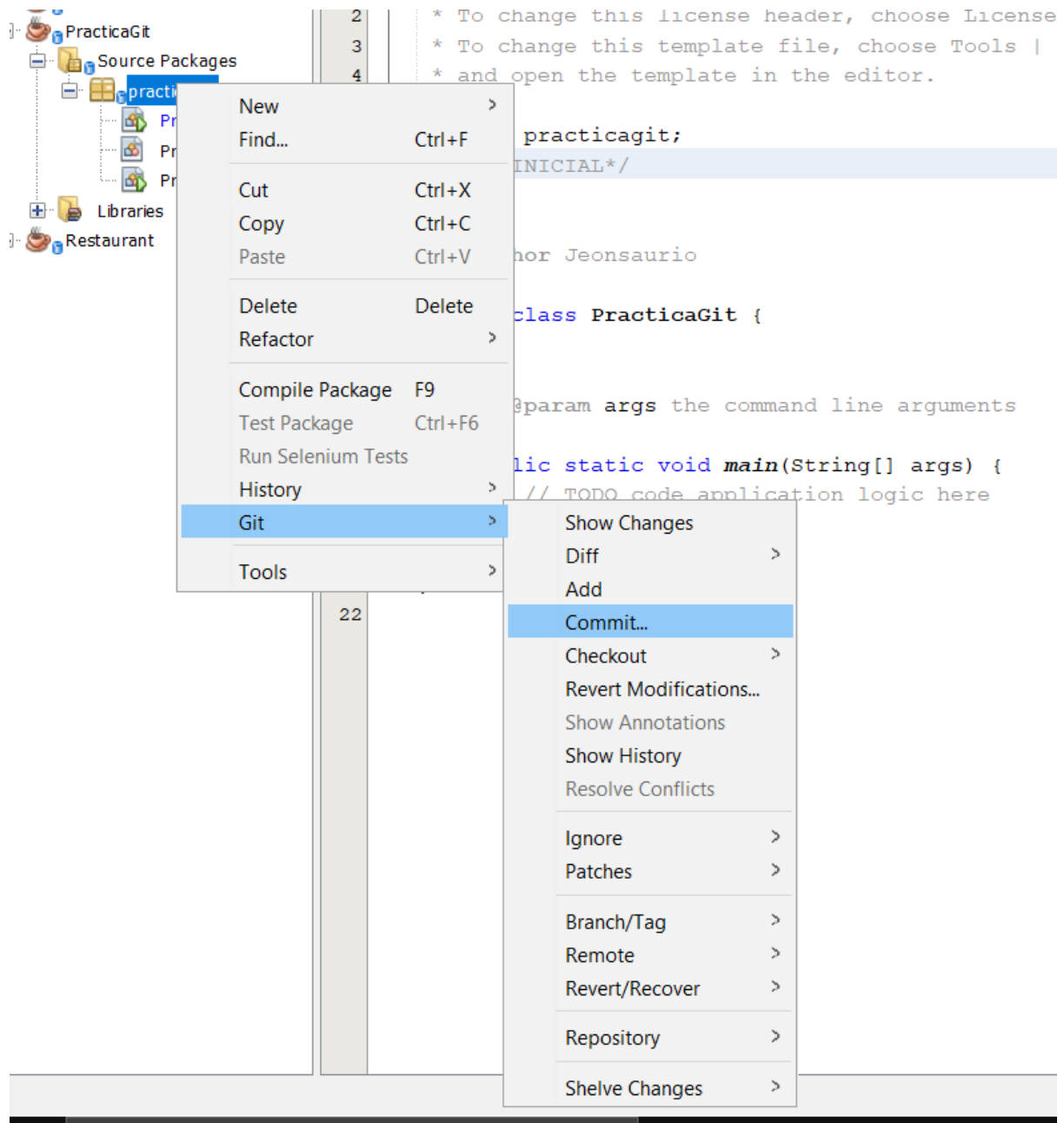
    */
    package practicagit;

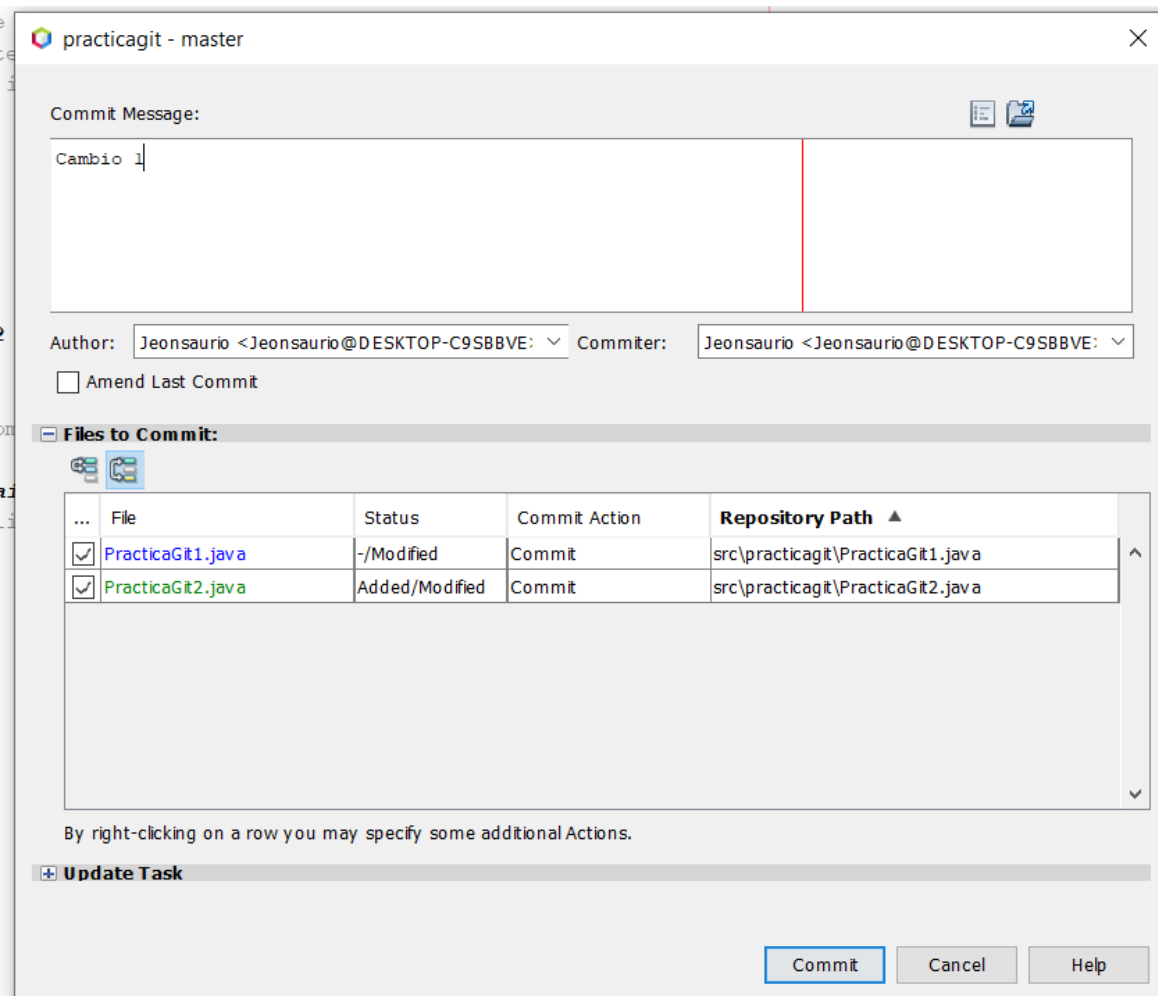
    /**
     *
     * @author Jeonsaurio
     */
    public class PracticaGit1 {
        /*HOLA*/
        /**
         * @param args the command line arguments
         */
        public static void main(String[] args) {
            // TODO code application logic here
        }
    }

```

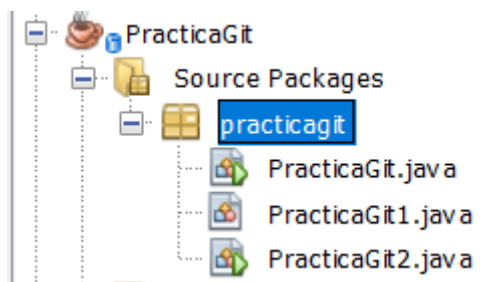


Para que los cambios vuelvan a guardarse y no salgan los archivos de color azul, hacemos el proceso del commit otra vez.





Hacemos el commit y nuevamente nuestros archivos volverán a ser de color negro.



Ahora, para subir el contenido de NetBeans al GitLab, entraremos a nuestra cuenta de Git y haremos un nuevo proyecto.. (Opción a la derecha una vez inicias sesión en el Git).

New project


Filter by name...

Name

★ 0 Y 0 I 0 D 0

Updated 3 weeks ago

El proyecto debe ser un “Blank Project”.



Create blank project

Create a blank project to house your files, plan your work, and collaborate on code, among other things.

Le damos un nombre a nuestro proyecto, y si quieres hacerlo privado para tu uso propio puedes habilitar esa opción.

les, plan
nong

Project name

NetBeansGit

Project URL

http://git.copernic.cat/roldan.garcia.laura/

Project slug

netbeansgit

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)

Description format

Visibility Level ?

☒ Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

☐ Internal
The project can be accessed by any logged in user except external users.

☐ Public
The project can be accessed without any authentication.

Project Configuration

☒ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project Cancel

Ahora ya tendríamos creado nuestro proyecto en el GitLab.

Project 'NetBeansGit' was successfully created. ✕

NetBeansGit Project ID: 1550 Star 0 Fork 0

1 Commit 1 Branch 0 Tags 92 KB Files 92 KB Storage

main netbeansgit / + ▾ History Find file Web IDE ▾ Clone ▾

Initial commit
Laura Roldán García authored just now add3ef94

README Auto DevOps enabled Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster

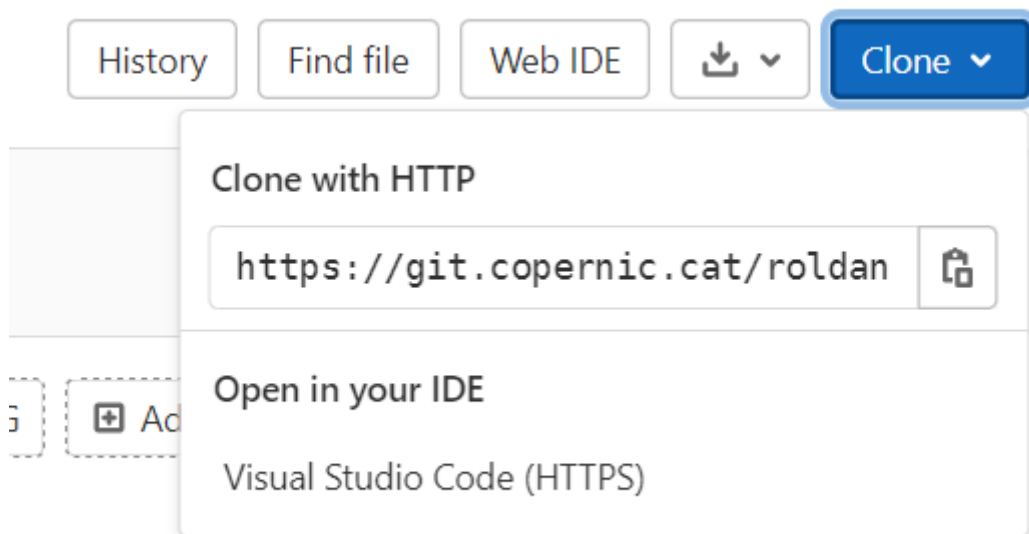
Configure Integrations

Name	Last commit	Last update
README.md	Initial commit	just now

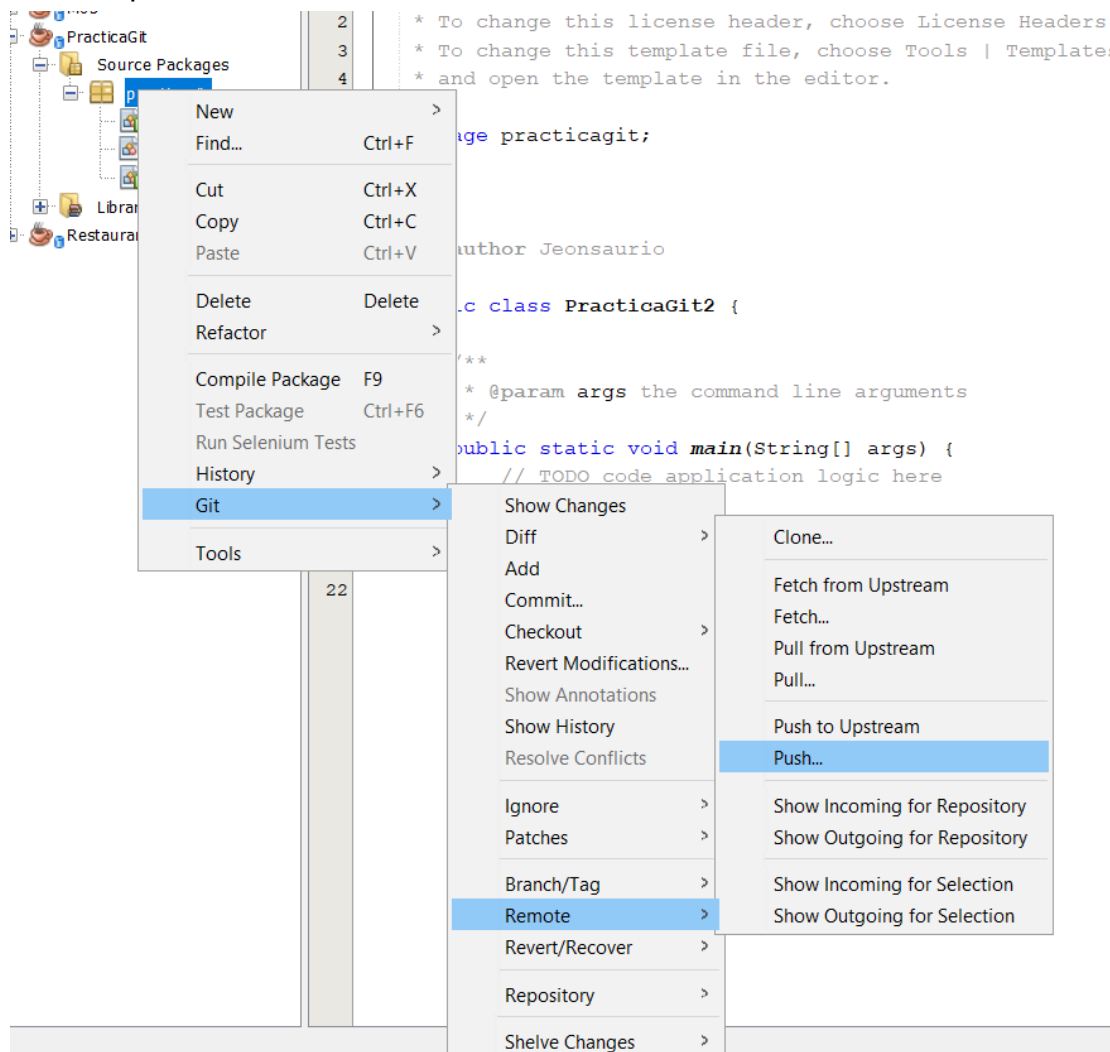
README.md

NetBeansGit

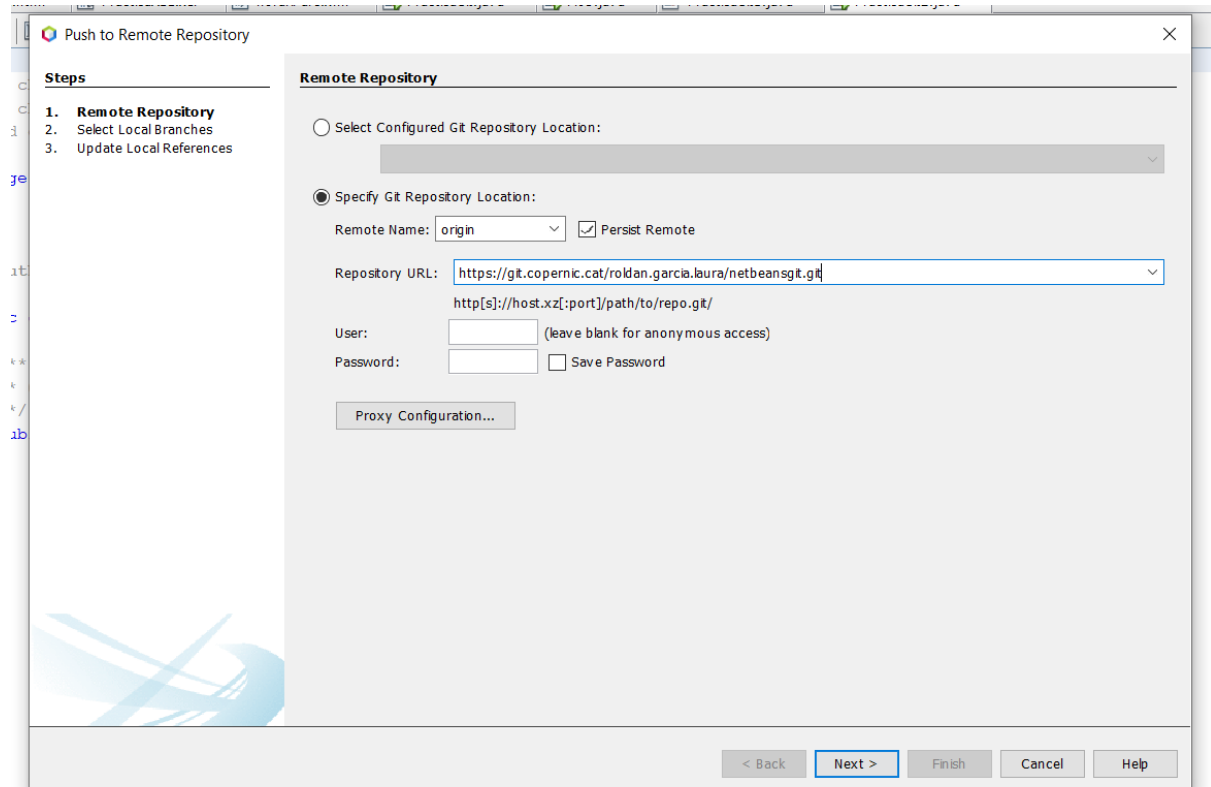
Y para poder conectar nuestro NetBeans con el Git, pulsamos el botón de Clone a la derecha y copiamos el link que nos proporciona.



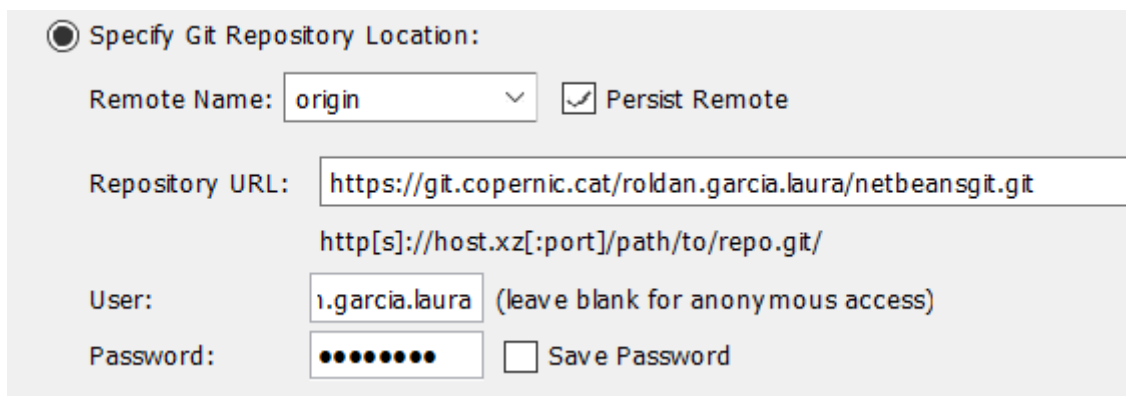
Volvemos a entrar al NetBeans, hacemos el mismo proceso de los pasos anteriores; darle click derecho al proyecto, opción de Git, y ahora vamos a la pestaña remote y le damos a push..



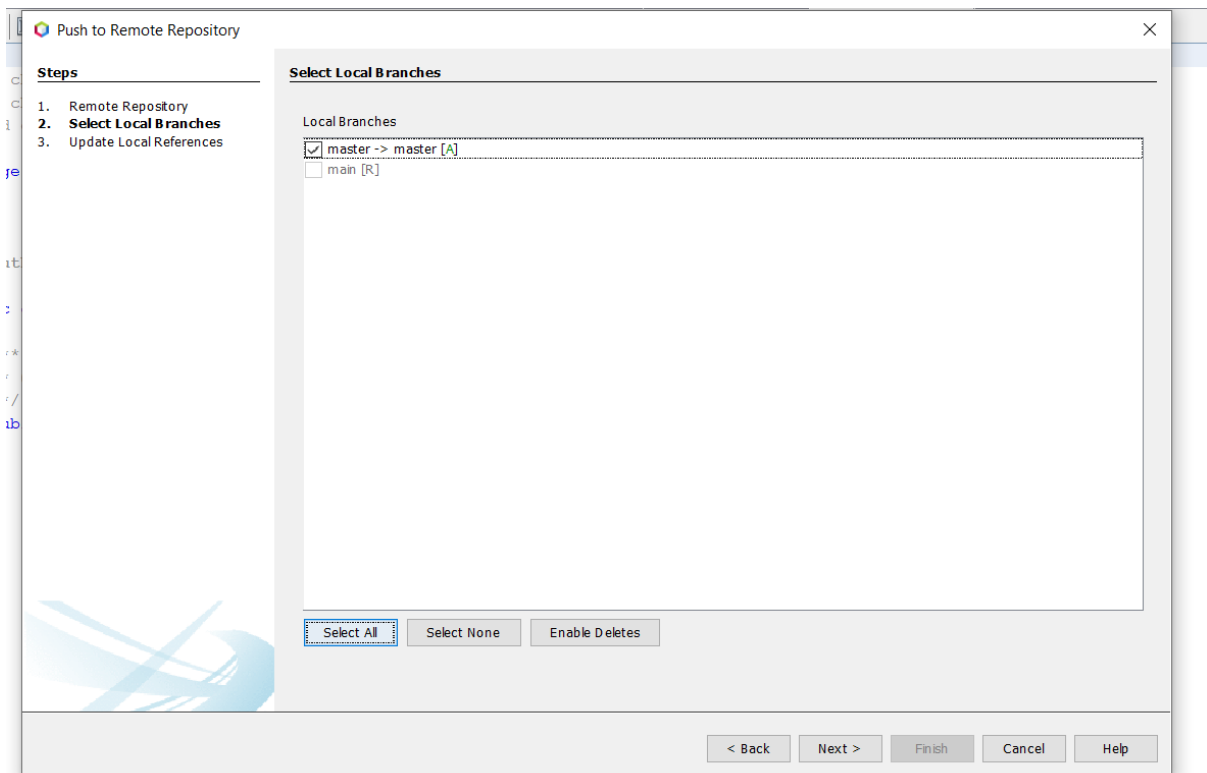
Ésto lo que hará es conectar el Git y el NetBeans. Añadiremos el link que nos ha proporcionado el Git a la opción de: Repository URL.



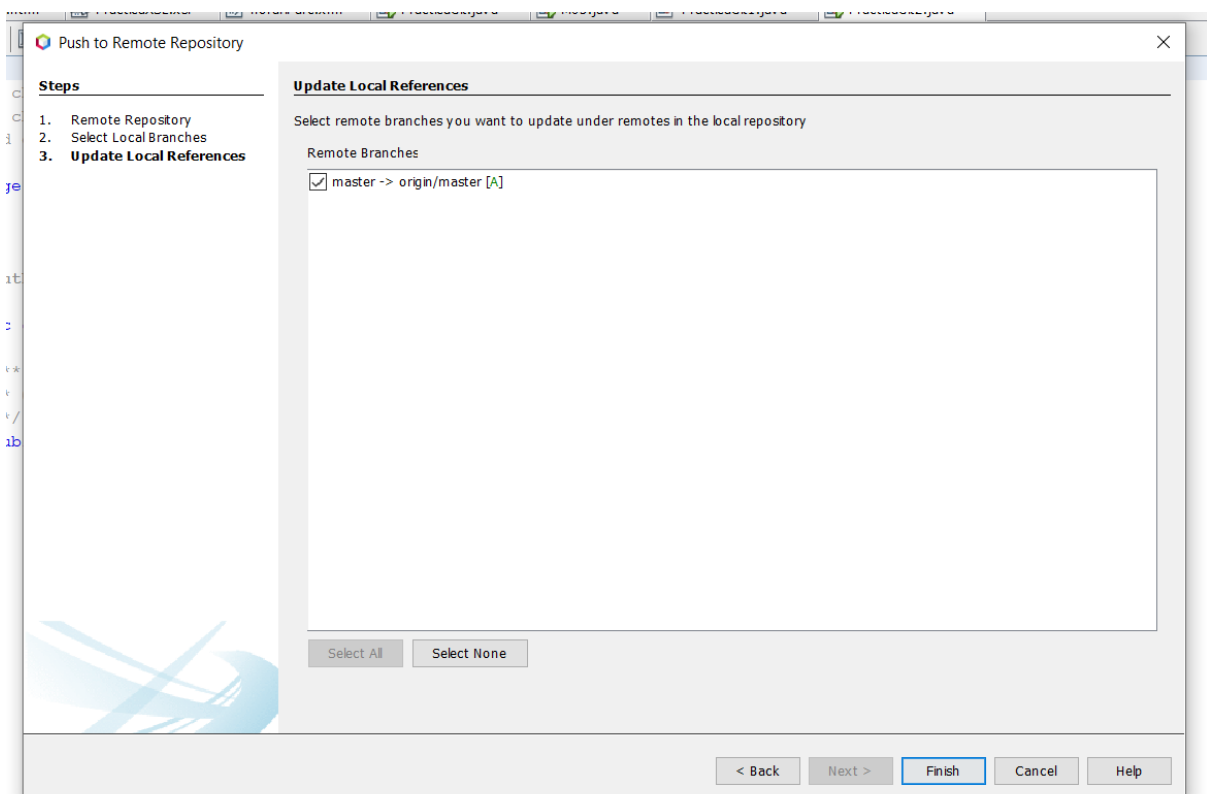
Y para que funcione ponemos nuestro usuario y contraseña. Finalmente le damos a next.



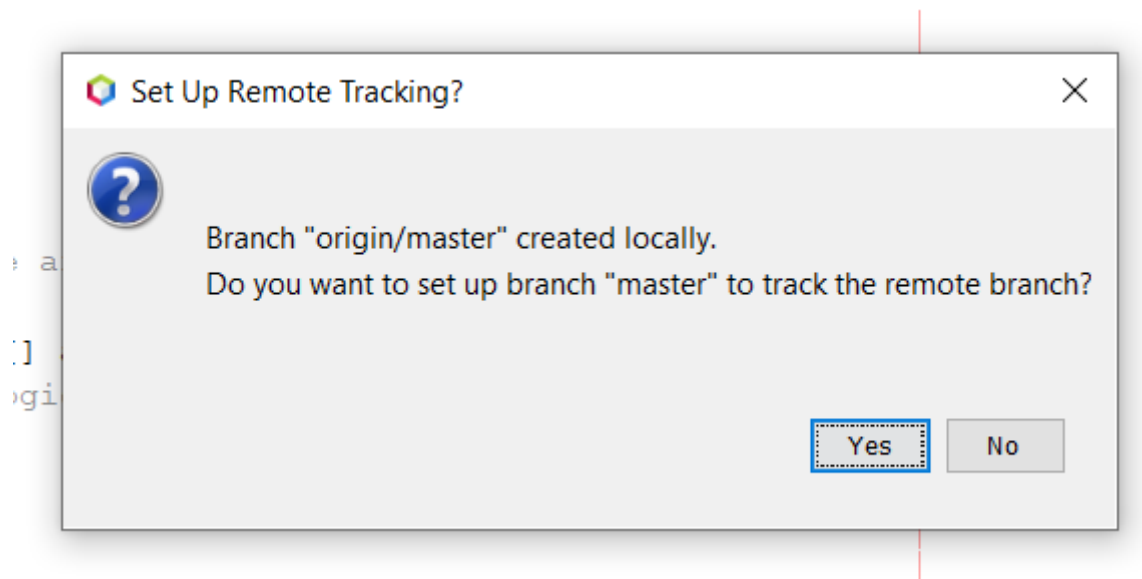
Seleccionamos el master y volvemos a darle a next.



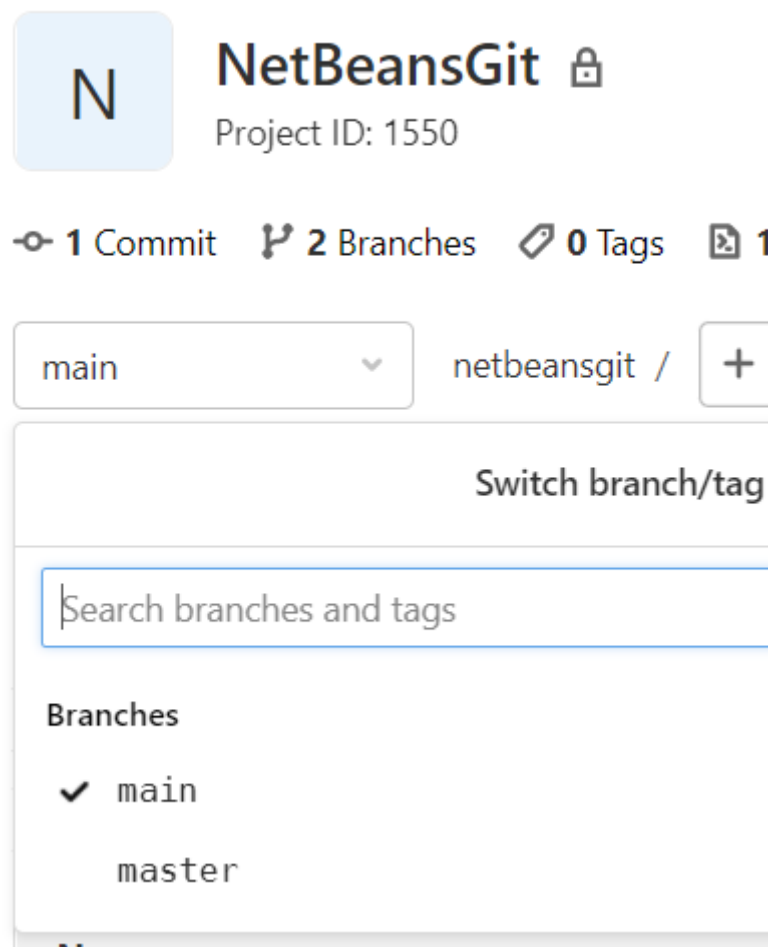
Finalmente le damos a Finish para acabar la unión.



Opción "Yes".



Vamos al Git y podemos comprobar que se ha realizado correctamente la unión.



master

▼

netbeansgit / src /

+

▼

Lock

History

Find file

Web IDE

▼

Clone ▼

Cambio 1

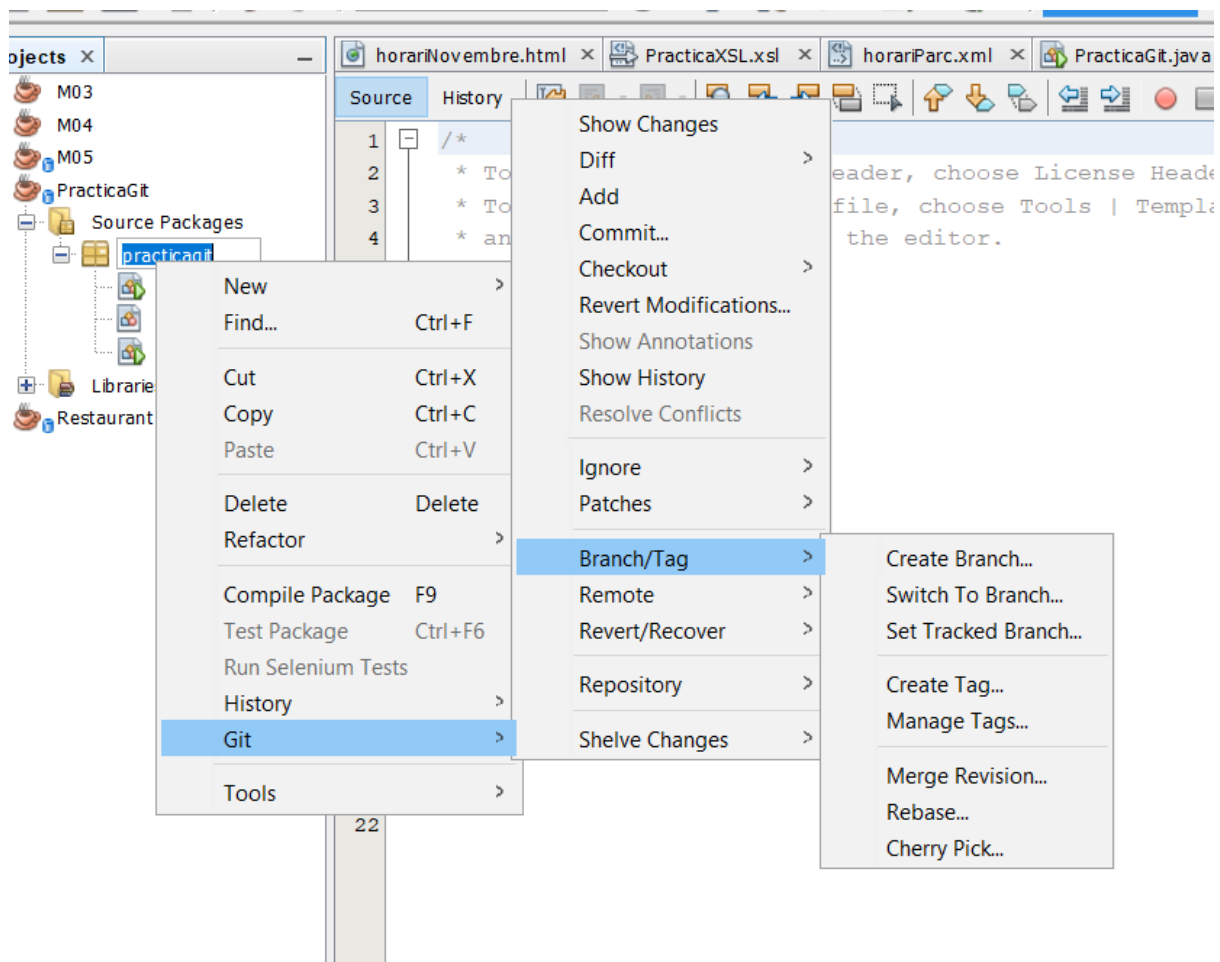
40a99c16

Jeonsaurio authored 15 minutes ago

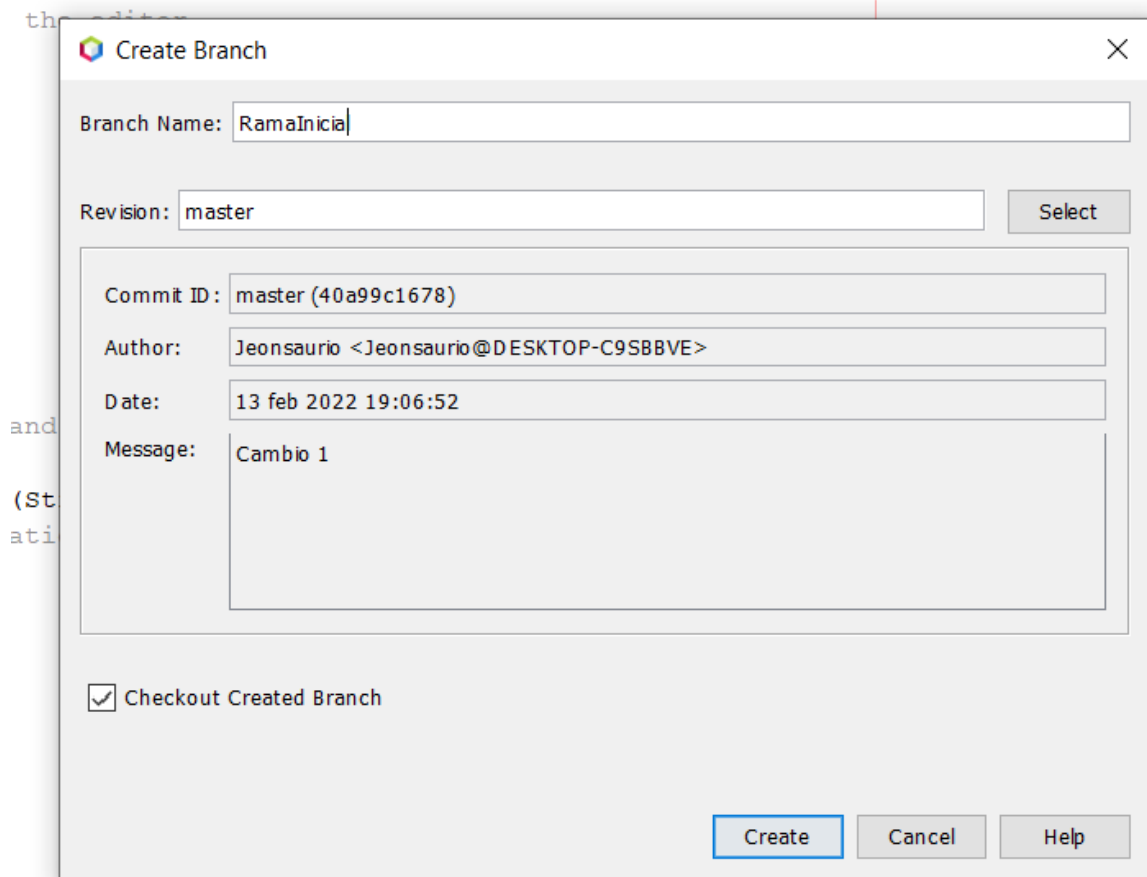
Name	Last commit	Last update
..		
<div></div> <div>practicagit</div>	Cambio 1	15 minutes ago

Ahora pasaremos al branching.

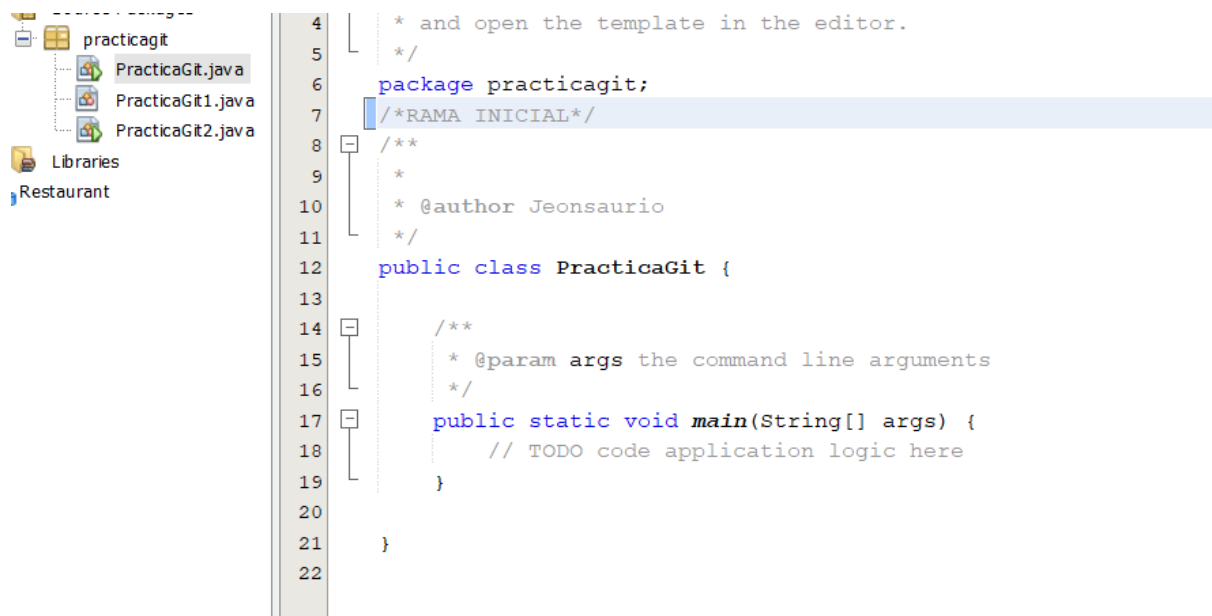
Para hacer una rama dentro del NetBeans, le daremos click derecho al paquete y entraremos en la pestaña de Git, dentro de ella la opción de Branch/Tag y crearemos una rama con el “Create Branch”.



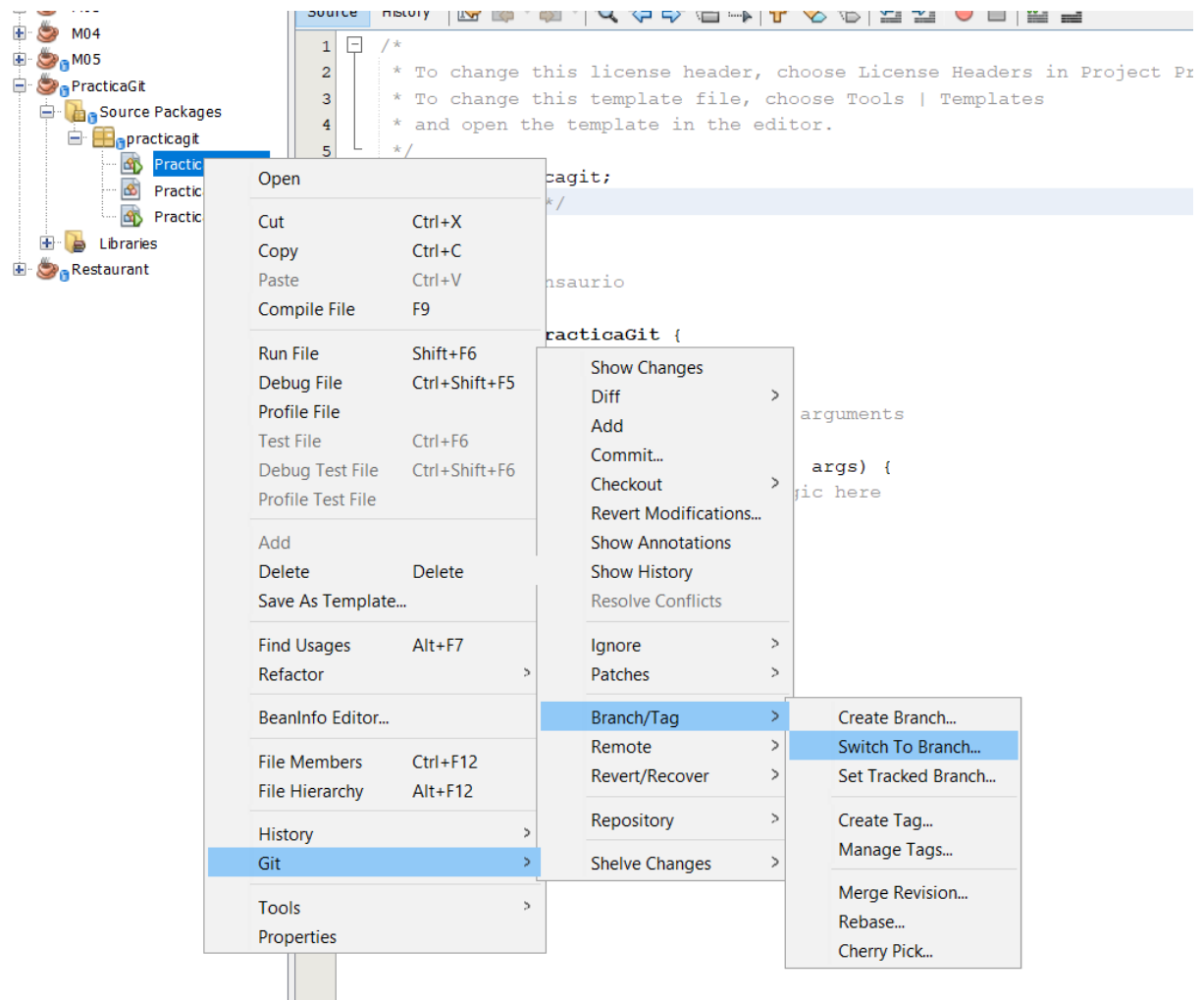
Le ponemos un nombre a la rama y le damos a Create.



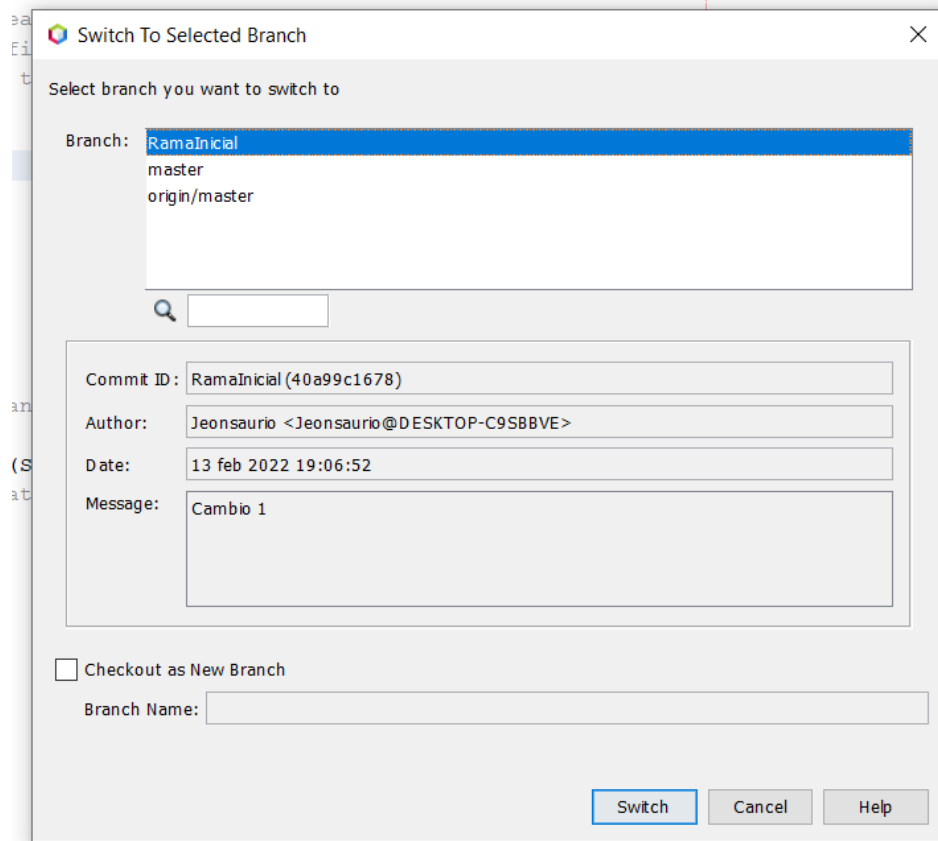
Ahora que la rama ya está creada, modificamos un poco el documento e intentamos cambiar de rama.



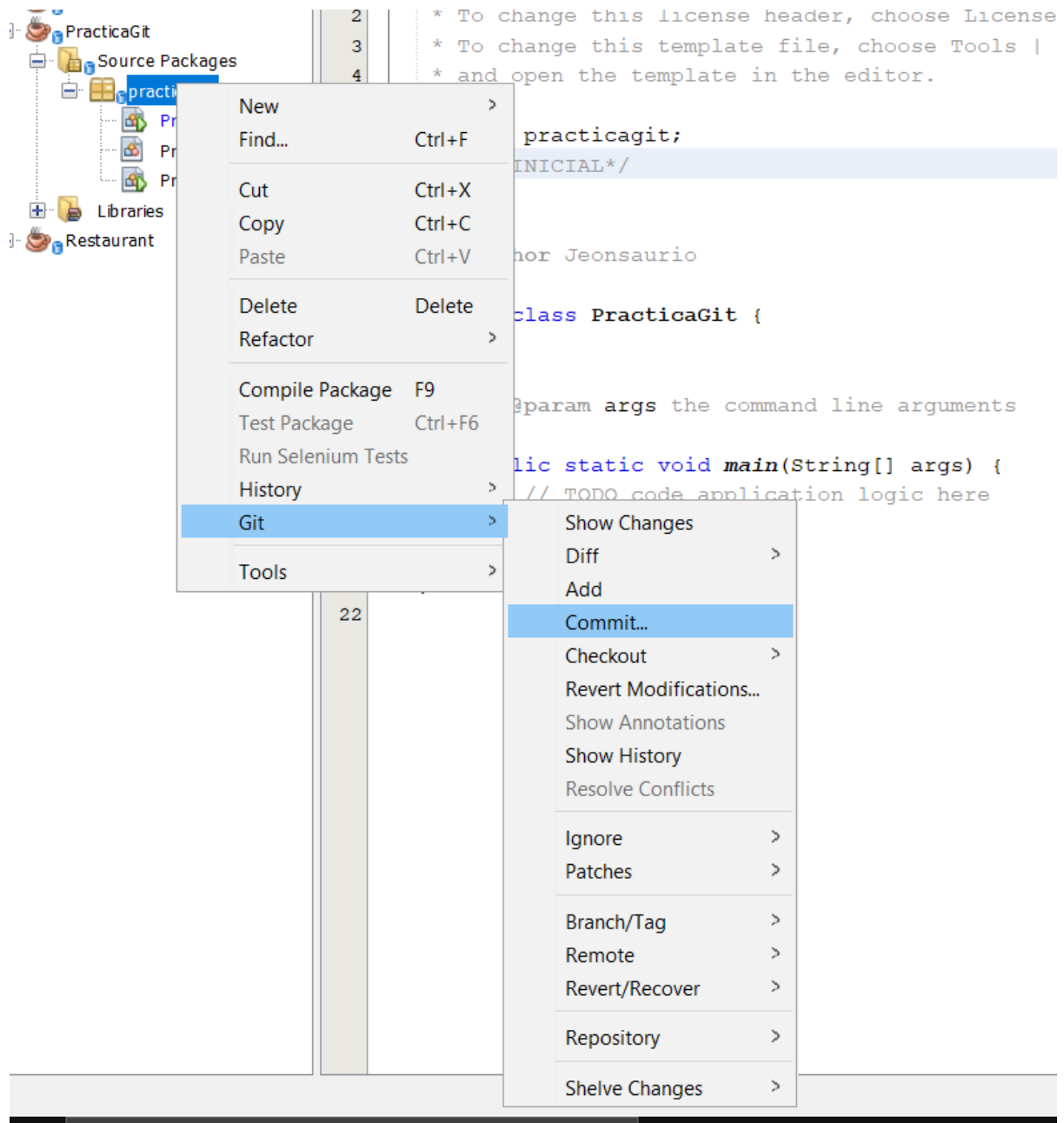
Para cambiar de rama, volvemos a darle click derecho al paquete, opción de Git, Branch/Tag y Switch to Branch.

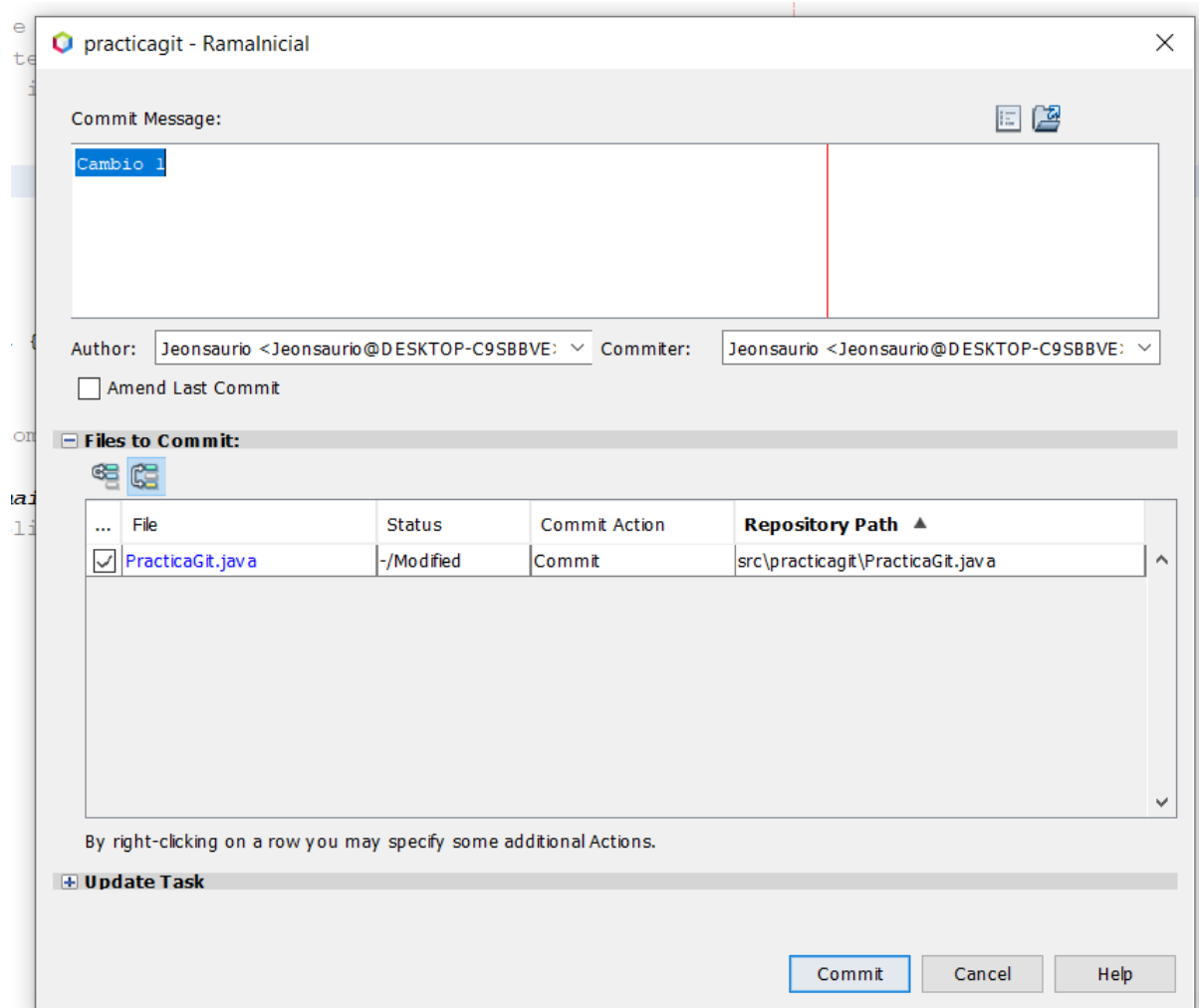


Ahora seleccionamos la rama que acabamos de crear y le damos al botón de Switch para continuar.

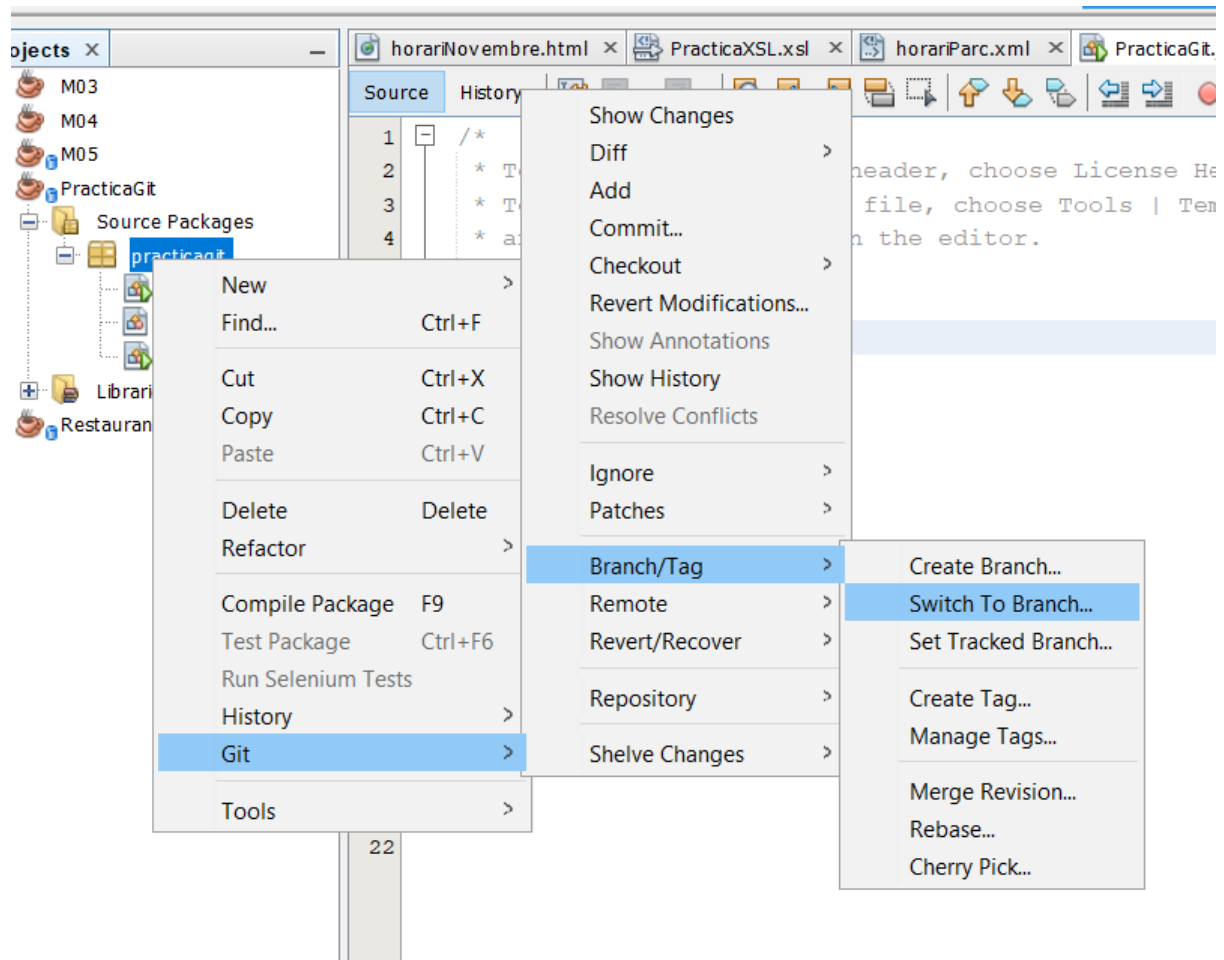


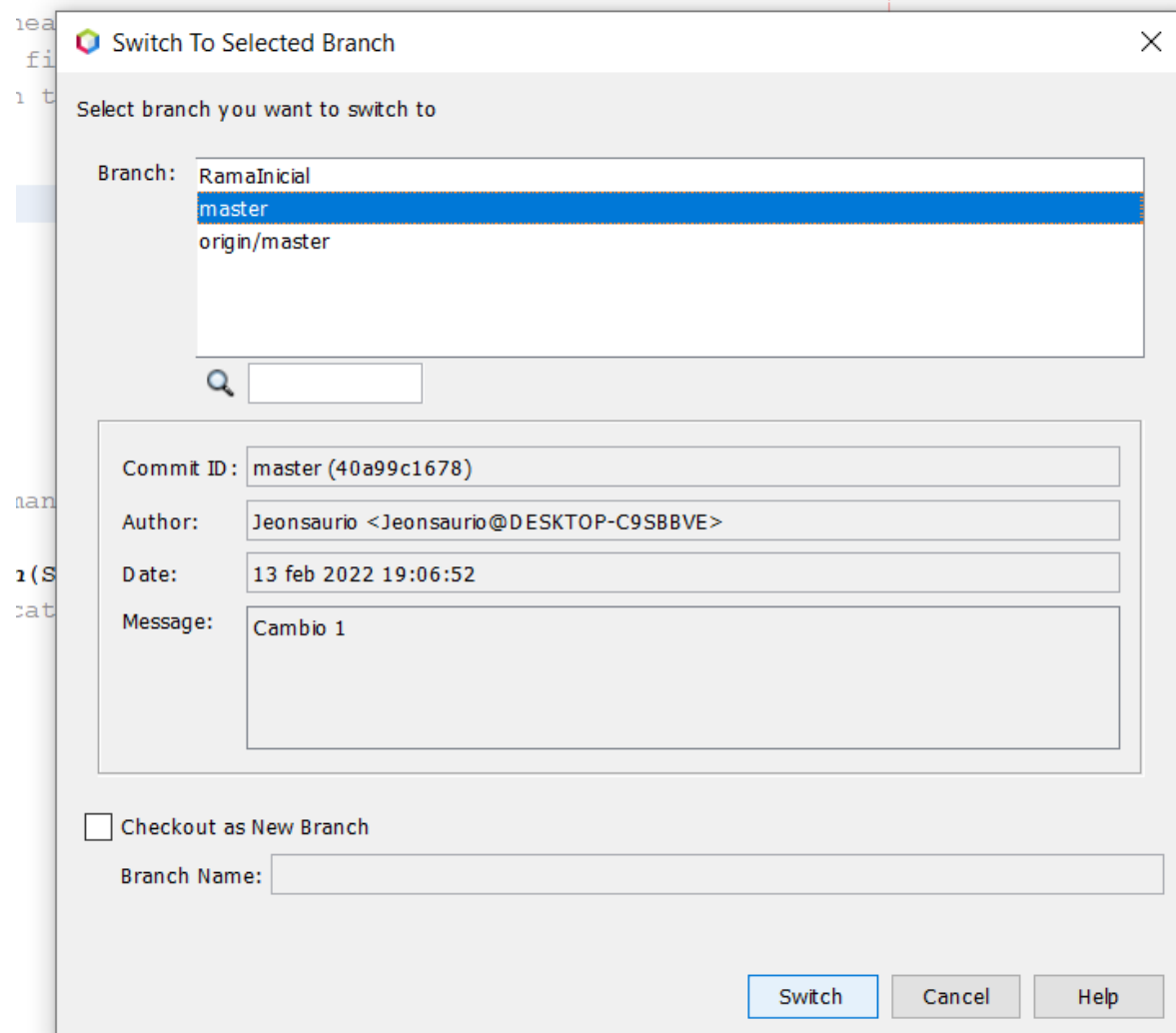
Podemos ver, que no habrá hecho ningún cambio, ya que necesitamos hacer un commit antes de nada, porque sino no nos dejará.



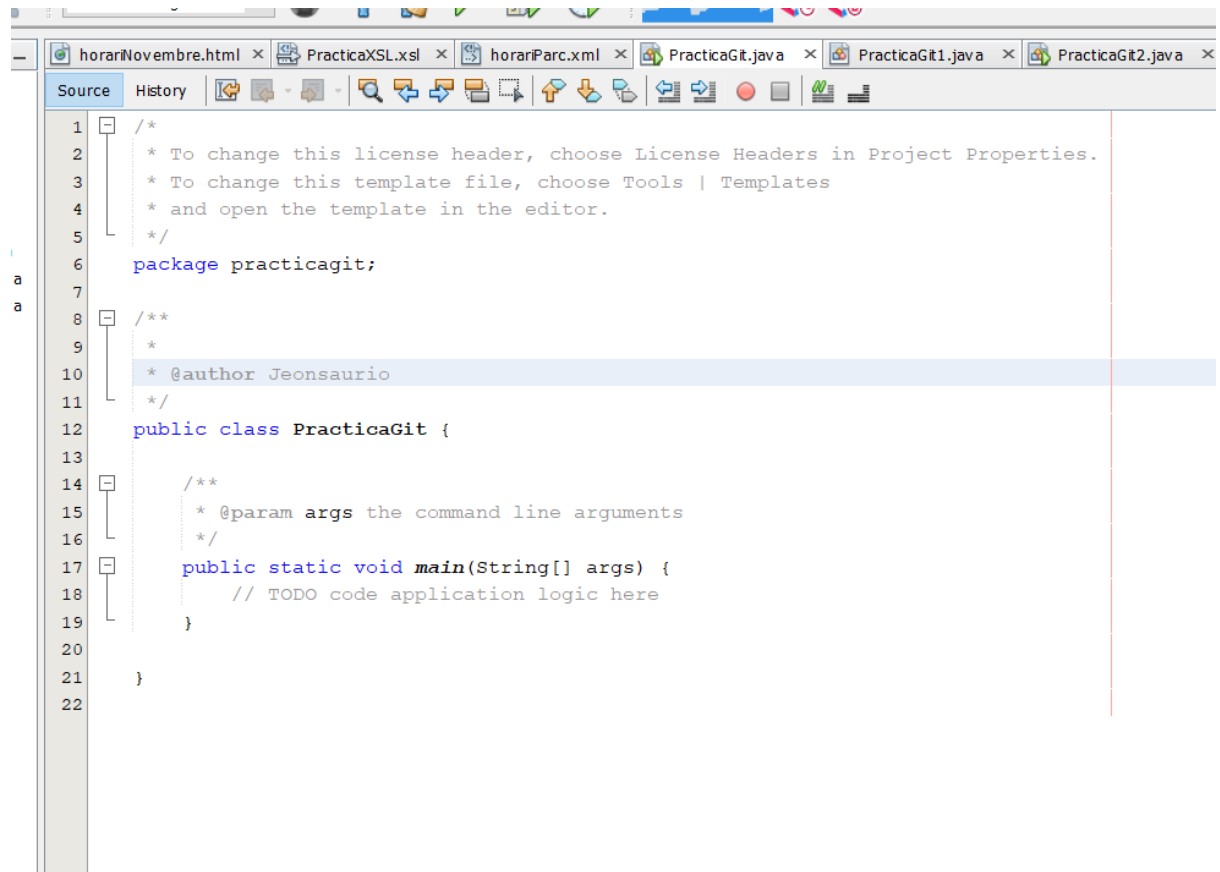


Después de guardar los cambios con el Commit, hacemos el mismo paso de antes, Switch to branch...



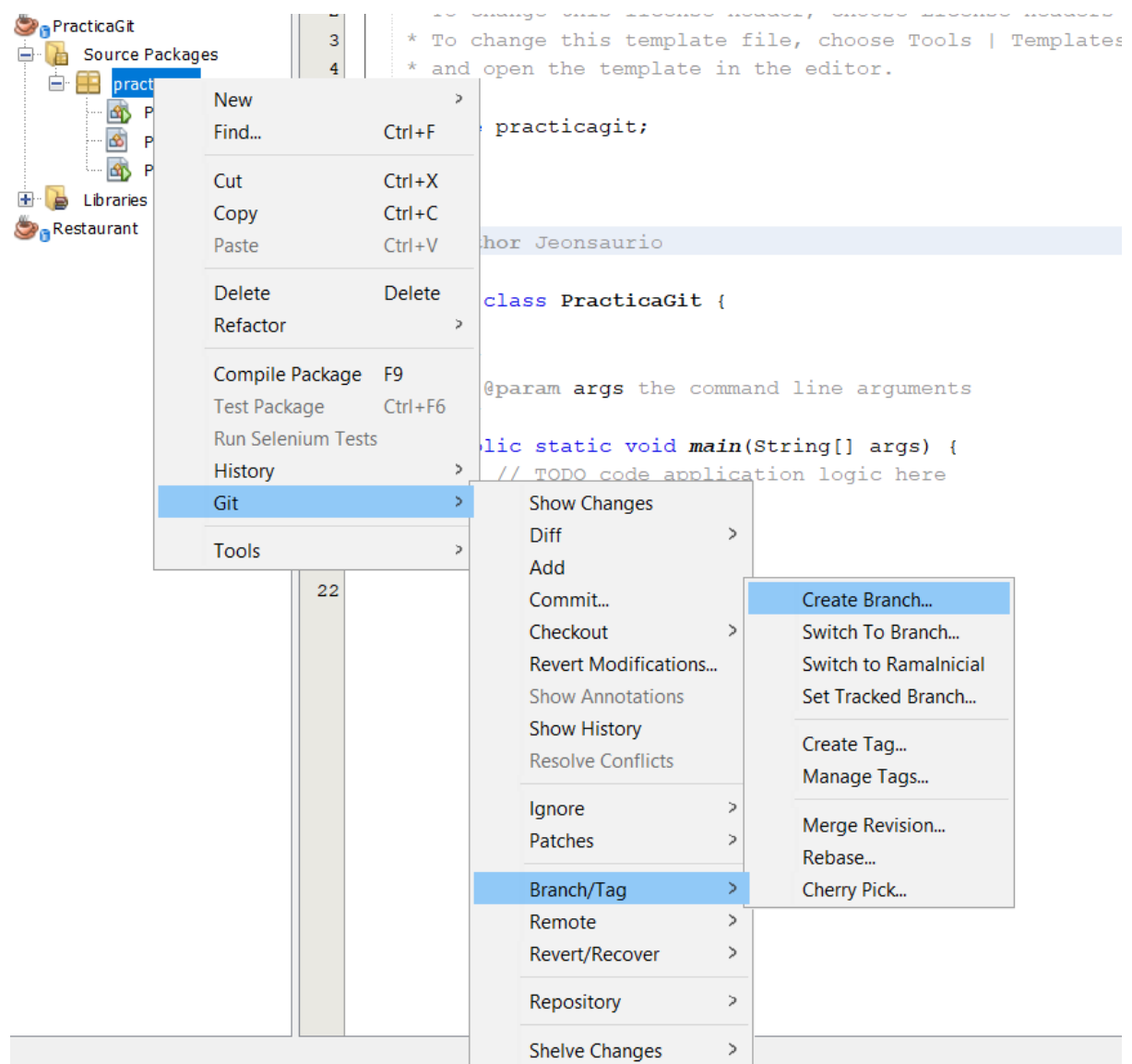


Y ahora podemos ver que ha vuelto a su estado principal.



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package practicagit;
7
8  /**
9   *
10  * @author Jeonsaurio
11  */
12  public class PracticaGit {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20
21  }
22
```

También puedes hacer una rama a partir de un commit en vez del archivo inicial.



En la pestaña de Revisión vemos que hay un botón de Select y seleccionamos el commit que queramos.

220, 0000 2022, 2022000
n the

nand
n (St
nati

Create Branch

Branch Name:

Revision:

Commit ID: master (40a99c1678)

Author: Jeonsaurio <Jeonsaurio@DESKTOP-C9S8BVE>

Date: 13 feb 2022 19:06:52

Message: Cambio 1

☒ Checkout Created Branch

Select Revision

Branches

- Local
 - RamaInicial
 - master**
- Remote
 - origin/master
- Tags

40a99c1 - Cambio 1
00c3df0 - Antes de hacer cambios

Show next: 10 | All | Refresh Filter:

Commit ID: 40a99c1678

Author: Jeonsaurio <Jeonsaurio@DESKTOP-C9S8BVE>

Date: 13 feb 2022 19:06:52

Message: Cambio 1

th editor

nd
St
ti

Create Branch

Branch Name:

Revision:

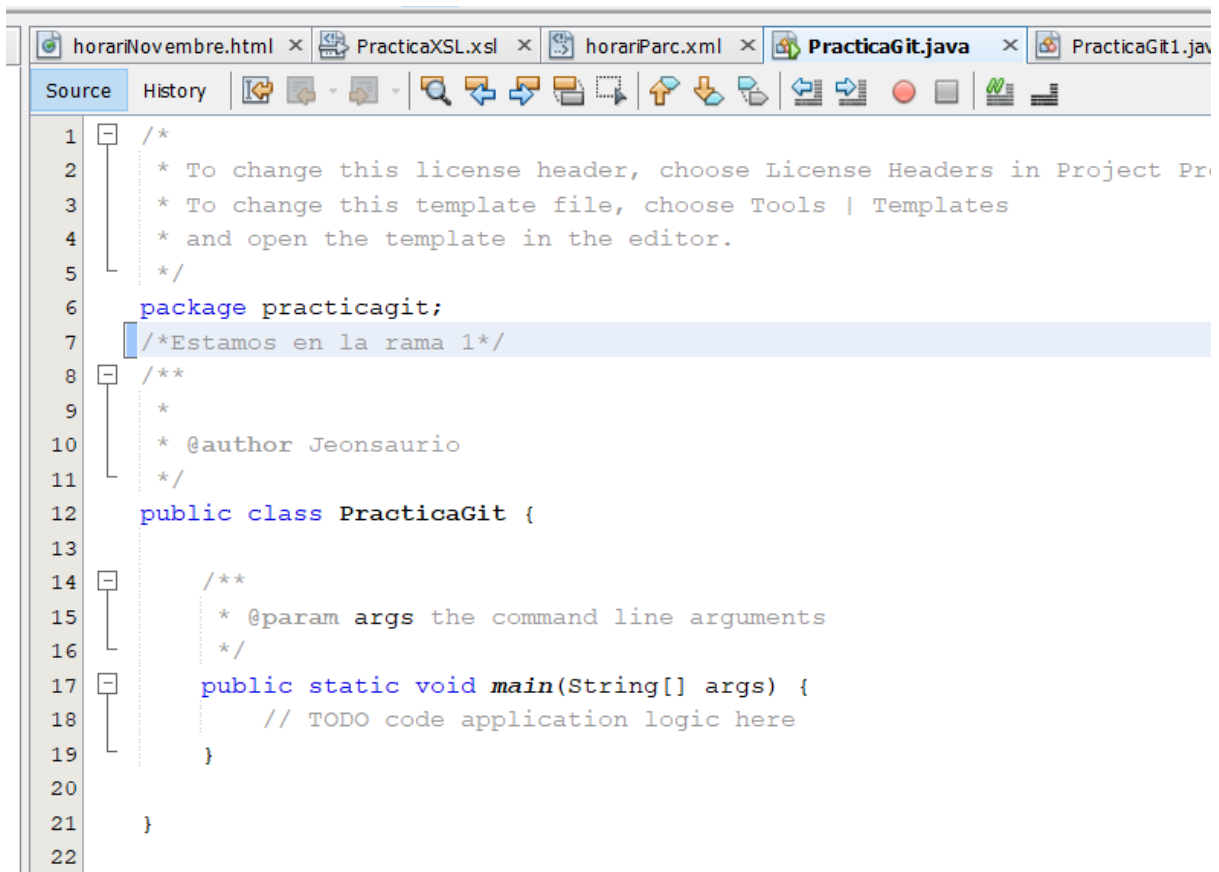
Commit ID: 40a99c1678

Author: Jeonsaurio <Jeonsaurio@DESKTOP-C9S8BVE>

Date: 13 feb 2022 19:06:52

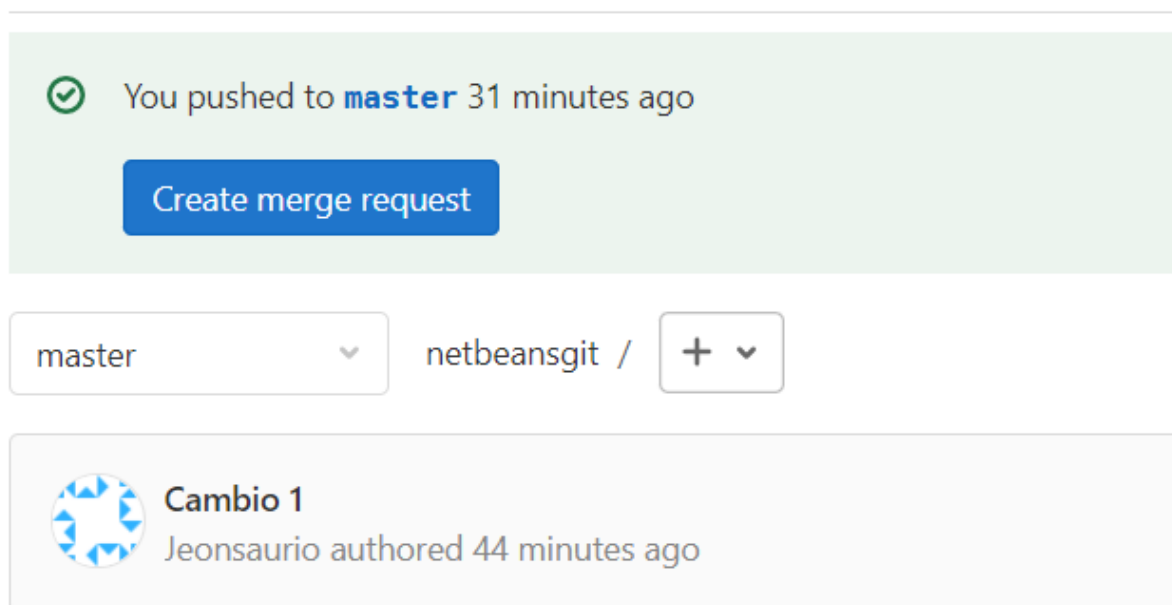
Message: Cambio 1

☒ Checkout Created Branch



```
1  /*
2   * To change this license header, choose License Headers in Project Properties
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package practicagit;
7  /*Estamos en la rama 1*/
8  /**
9   *
10   * @author Jeonsaurio
11   */
12  public class PracticaGit {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20
21  }
22
```

Volvemos al GIT para implementar los cambios de las ramas al archivo principal. Vemos que nos sale un mensaje de Create merge request, le damos



Dejamos las opciones por defecto y lo creamos.

[Open](#) Created just now by  **Laura Roldán García** [Maintainer](#)

[Edit](#) [Mark as draft](#) [▼](#)

master


[Overview](#) [0](#) [Commits](#) [2](#) [Changes](#) [3](#)

 **Request to merge `master` into `main`**
The source branch is [1 commit behind](#) the target branch

[Open in Web IDE](#) [Check out branch](#) [⌵](#)

 Approval is optional

[View eligible approvers](#)

 [Merge](#) ☒ Delete source branch ☐ Squash commits [?](#)

[2 commits](#) and [1 merge commit](#) will be added to `main`. [Modify merge commit](#)

 0  0 

[Oldest first](#) [Show all activity](#)

[Write](#) [Preview](#)

B *I* “ ” </> [🔗](#) [☰](#) [☷](#) [☰](#) [☷](#) [📎](#) [📁](#) [📄](#)

Write a comment or drag your files here...

Y ahora todas las ramas creadas se han juntado en un único archivo. Ésto nos sería útil en una empresa, donde varias personas tienen que trabajar sobre una misma base y luego juntarlos todos.