



# PRÁCTICA

## DESCRIPCIÓN

Podemos ver la realización esta práctica del git desde el IntelliJ

Daniel Santiago Rodríguez Sierra

## PRÁCTICA GIT

### Contenido

PRÁCTICA GIT .....	2
R01. Creación del repositorio.....	3
R03. Hello World .....	4
R04. Modificamos la calculadora .....	4
R05. Diferencias .....	5
R06. Nuevo fichero.....	6
R07. Renombramos el fichero.....	7
R08. Borramos un fichero .....	8
R09. ¡Recuperamos el fichero perdido! .....	8
R10. Ordenación.....	9
R11. Buscador.....	10
R12. Comparamos y fusionamos ramas.....	11
R13. Borramos las ramas sobrantes.....	15
R14. Subimos los cambios a Bitbucket.....	15
R15. Resolvemos problemas .....	17
R16. ¿Quién ha hecho que? .....	18

## R01. Creación del repositorio

Daniel Santiago Rodriguez / PracticaGit1 / Rodriguez\_Sierra\_Daniel / Repository settings


### User and group access

Add members

Grant access to this repository by adding users and [groups](#). You can find them by name if they already have workspace access. If not, type a full email address to add a new user. [Learn more about granting access to repositories.](#)

For a list of all users with access to any of your private repositories and to see which users count towards your bill, go to [Users on plan](#).

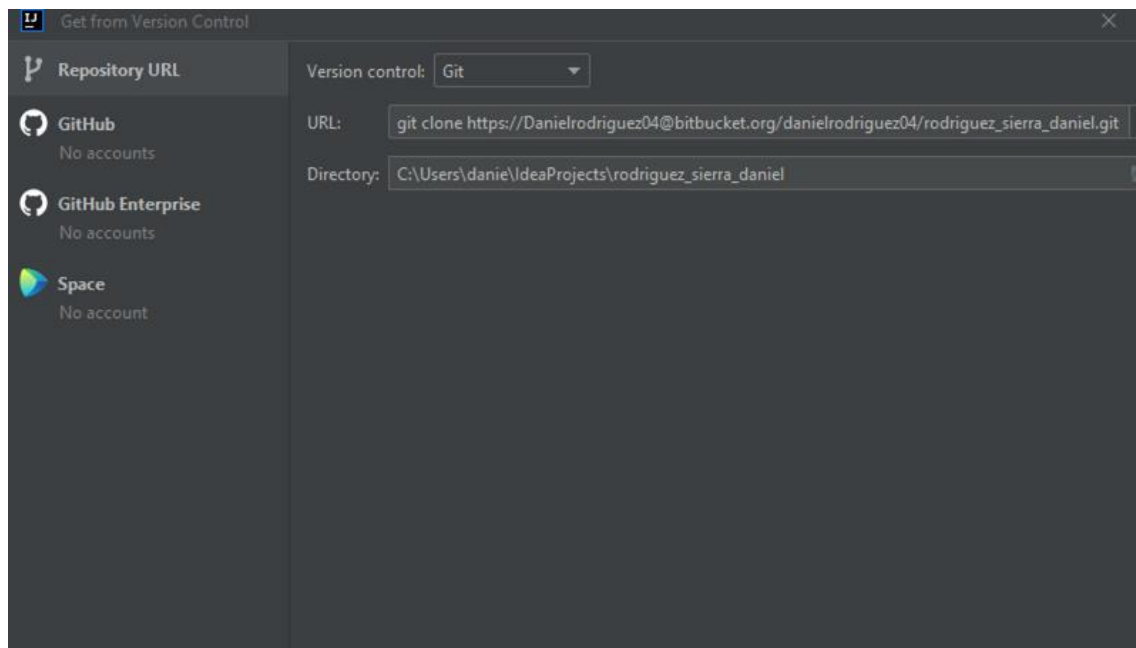
▾

<input type="checkbox"/>	Name	Permission	Actions
<input type="checkbox"/>	 <b>Joan Galmés</b> Joan Galmés	Admin ▾	<a href="#">Remove</a>

< **1** >

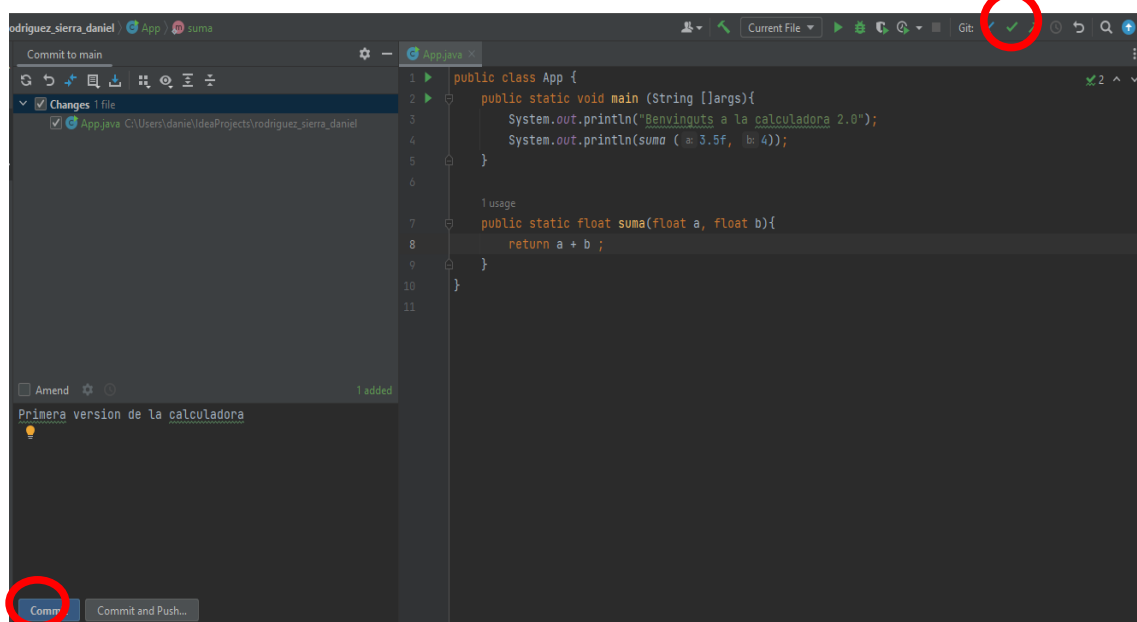
## R02. Clonación del repositorio

### Git clone

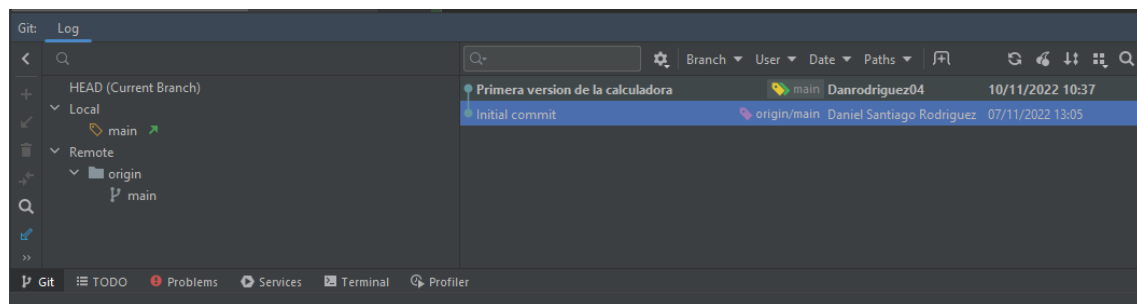


### R03. Hello World

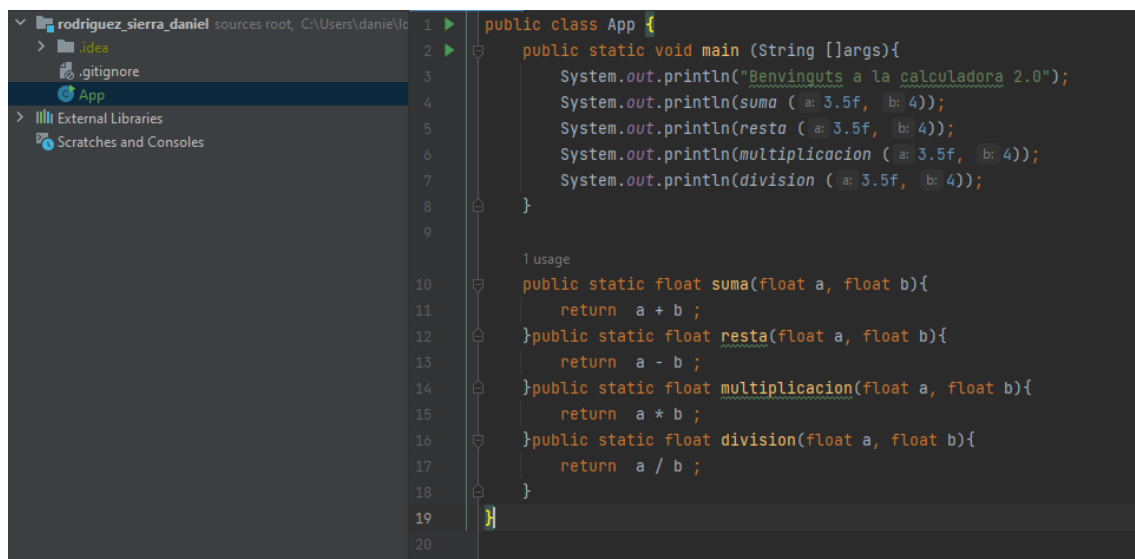
crear archivo java

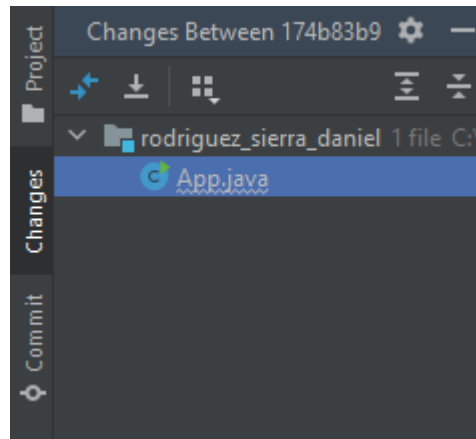


git add y git commit y para ver los commits un git log



### R04. Modificamos la calculadora

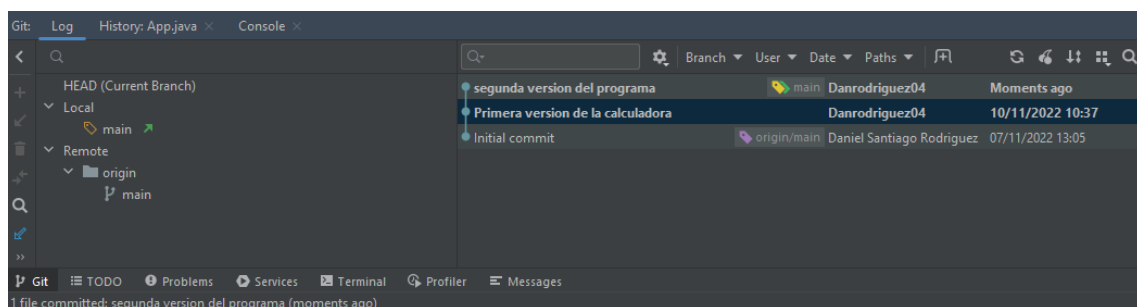
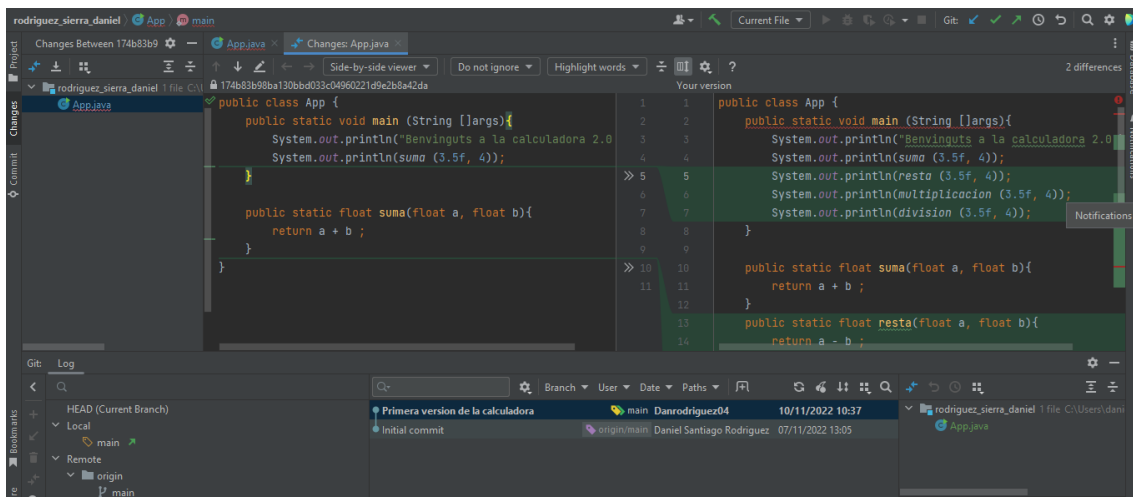




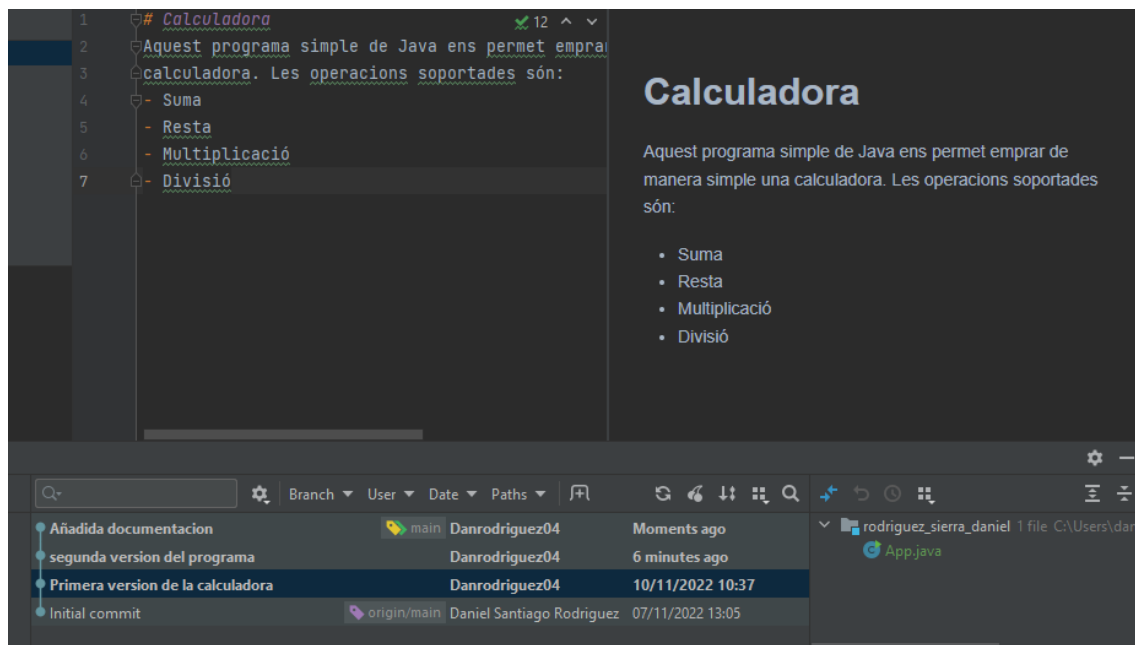
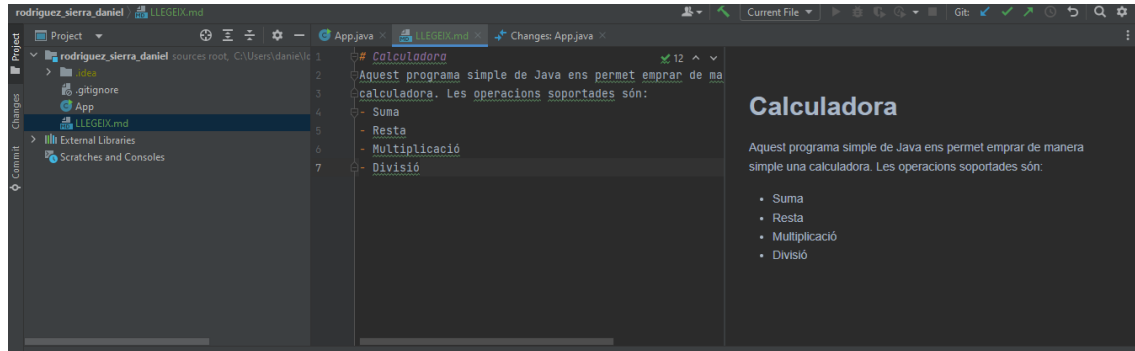
## Ro5. Diferencias

Git diff “versión del commit”

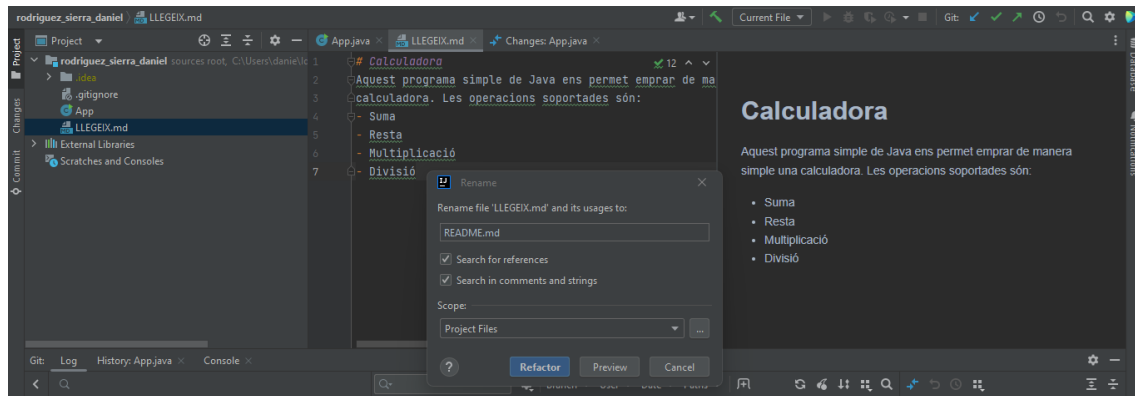
Las diferencias de los dos códigos son las que podemos ver que están resaltadas en verde



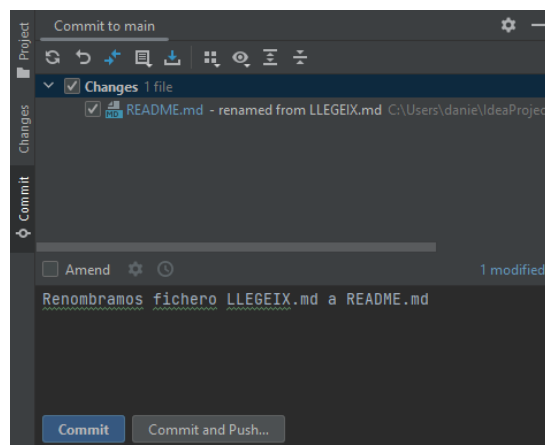
## Ro6. Nuevo fichero



## Ro7. Renombramos el fichero

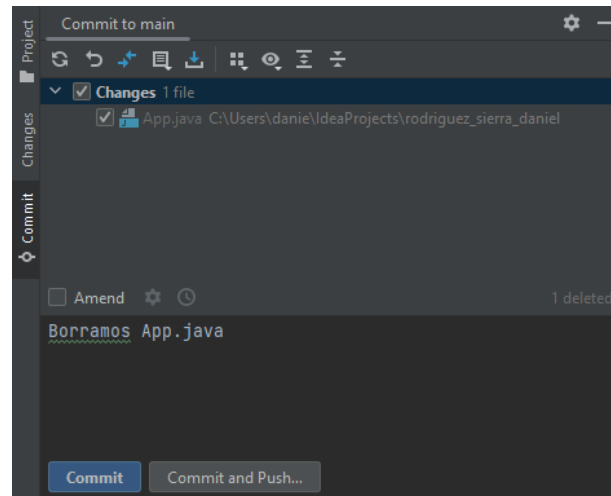


-Al hacer el commit IntelliJ muestra los cambios de los archivos que se subirán

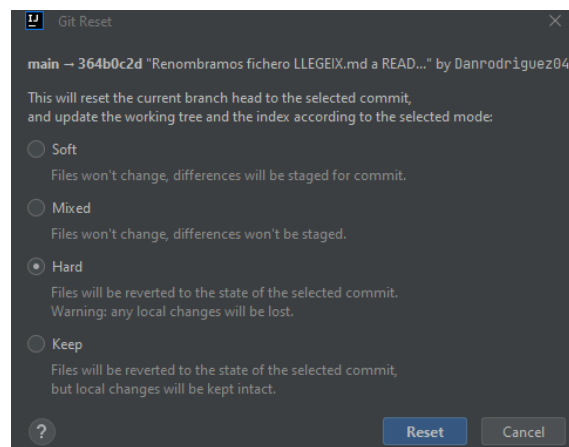
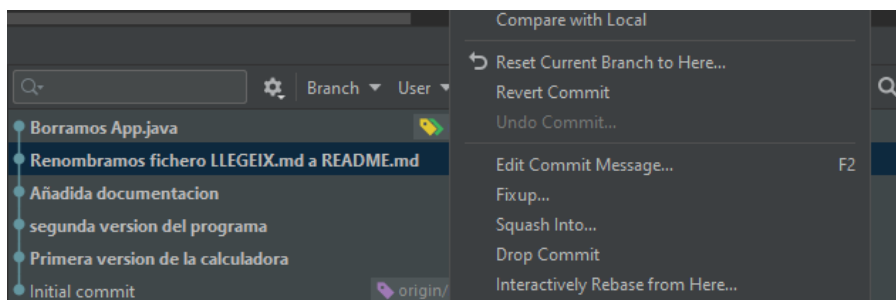


Renombramos fichero LLEGEIX.md a README.md	main	Danrodriguez04	Moments ago
Añadida documentacion		Danrodriguez04	Yesterday 15:09
segunda version del programa		Danrodriguez04	Yesterday 15:03
Primera version de la calculadora		Danrodriguez04	10/11/2022 10:37
Initial commit	origin/main	Daniel Santiago Rodriguez	07/11/2022 13:05

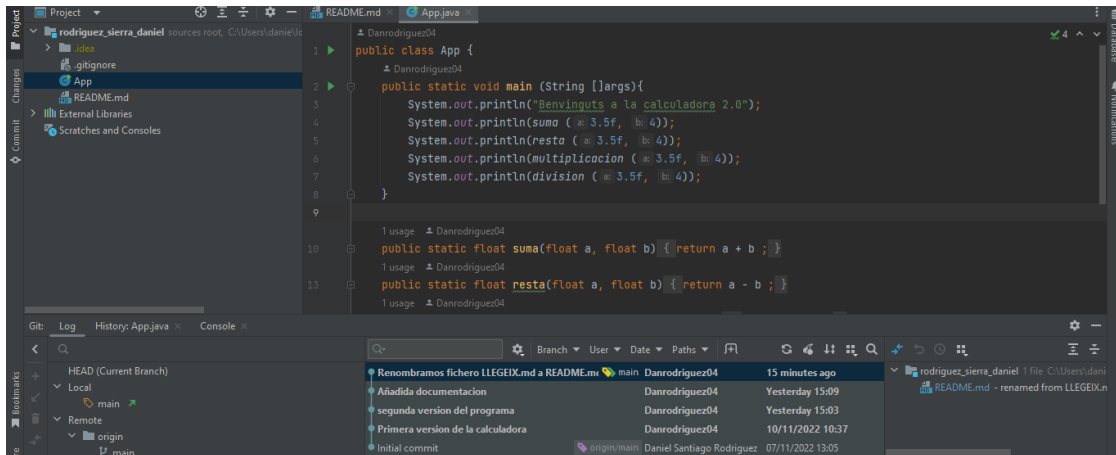
## Ro8. Borramos un fichero



## Ro9. ¡Recuperamos el fichero perdido! Utilizamos un Git reset hard

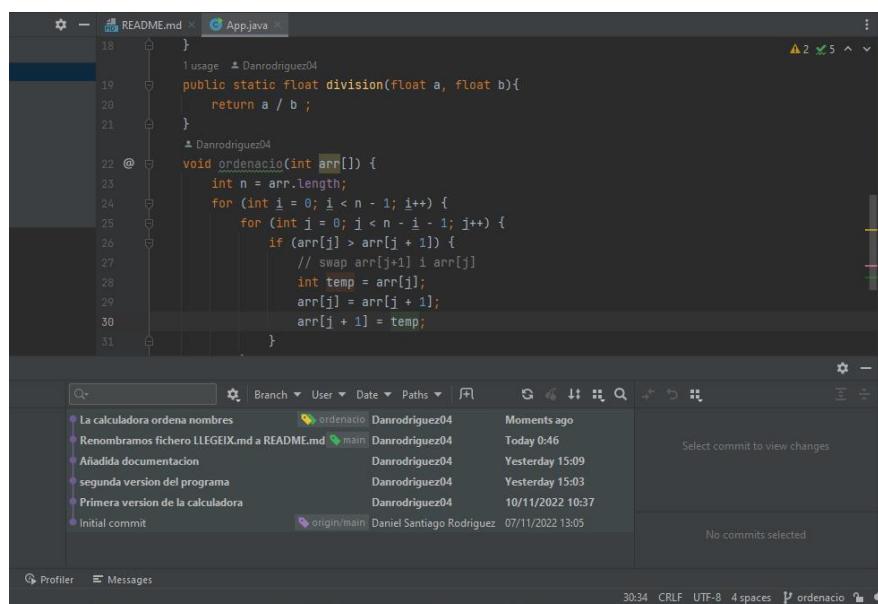
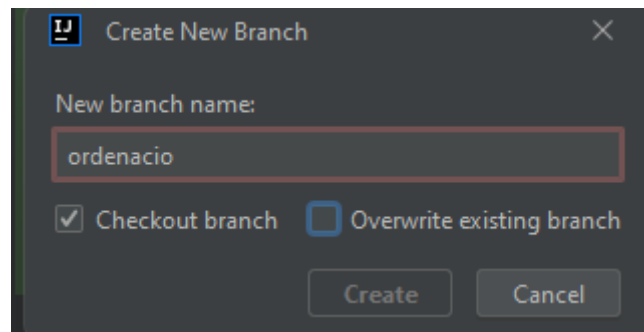


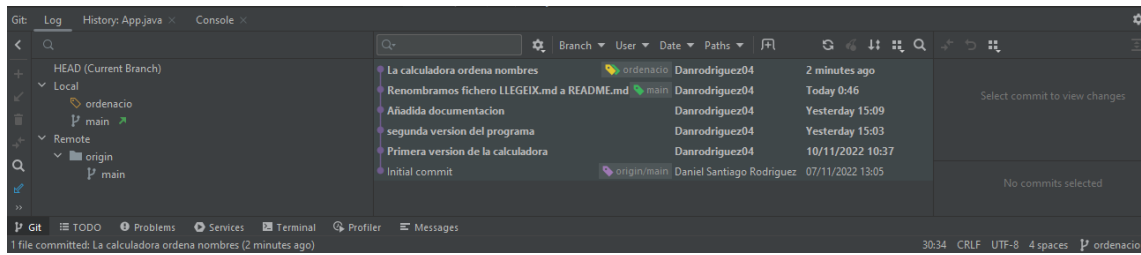




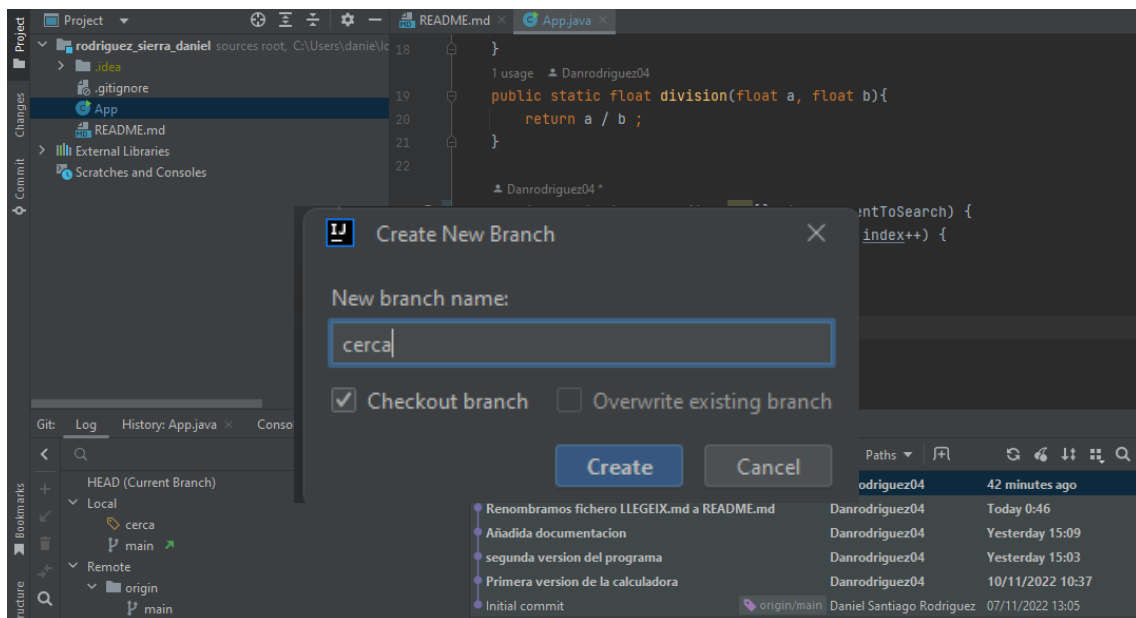
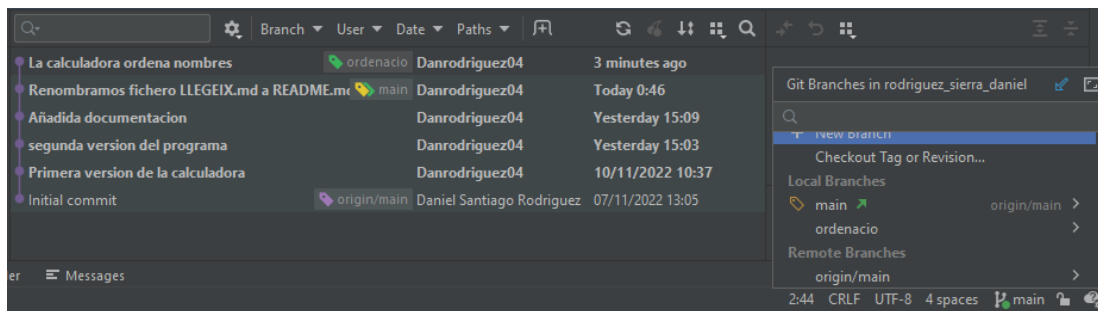
## R10. Ordenación

Git Branch “y el nombre de la rama ” y git checkout para cambiar de rama



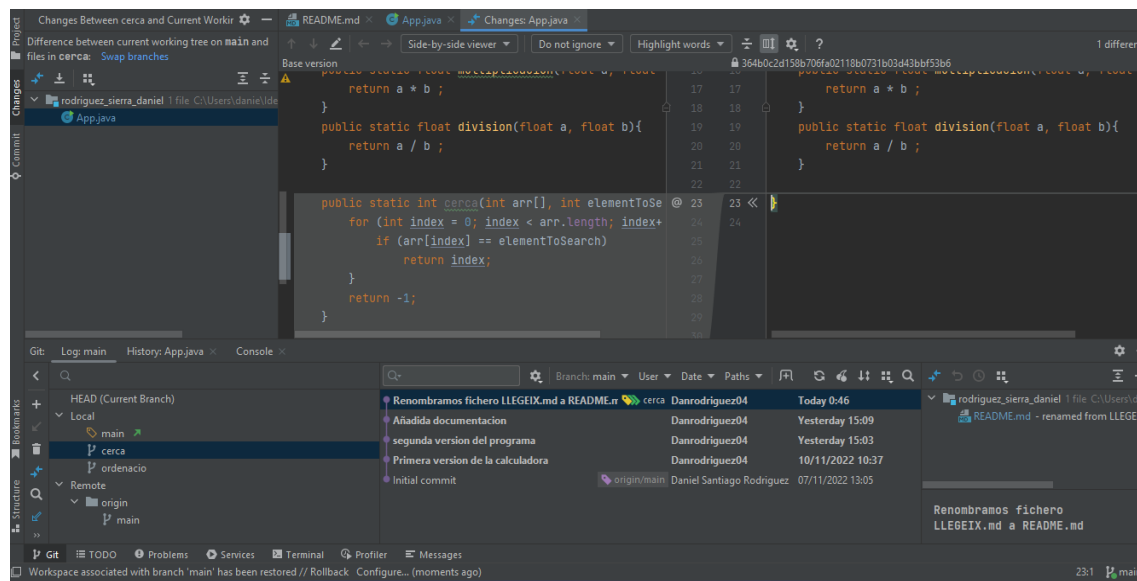
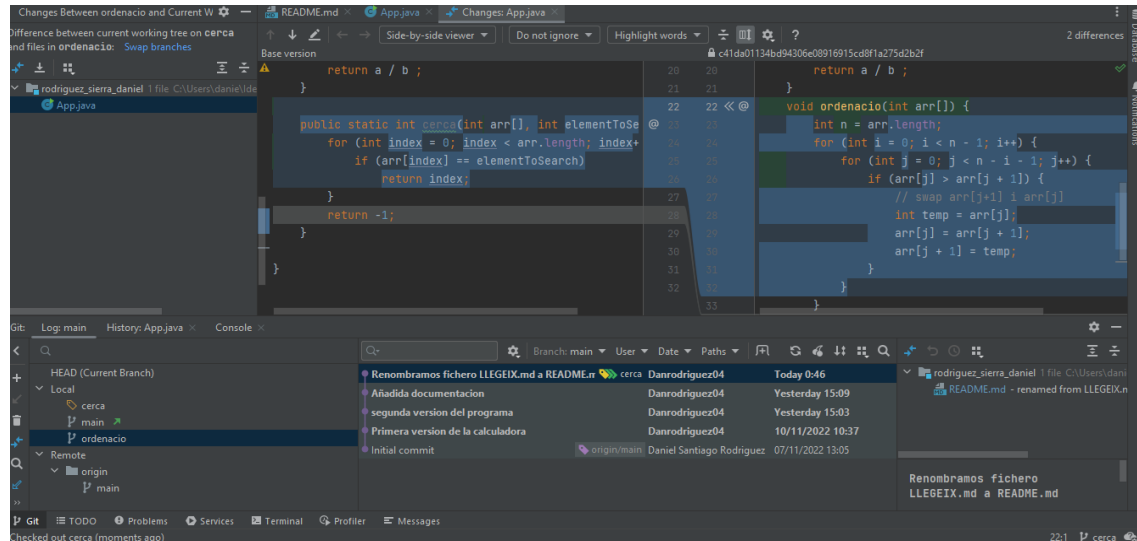


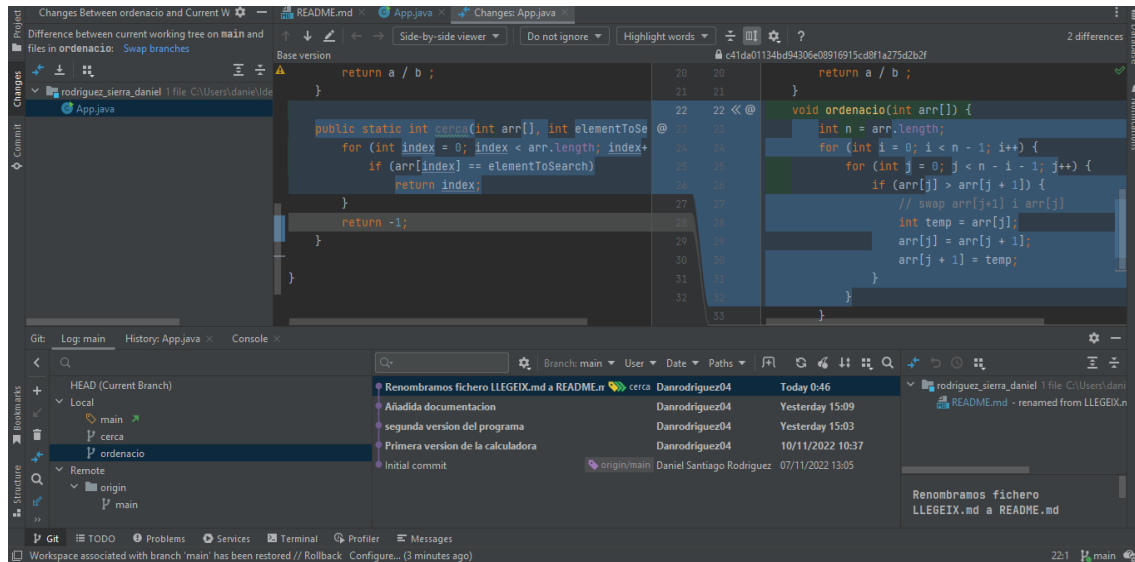
## R11. Buscador



## R12. Comparamos y fusionamos ramas

Compara el contenido de las tres ramas creadas

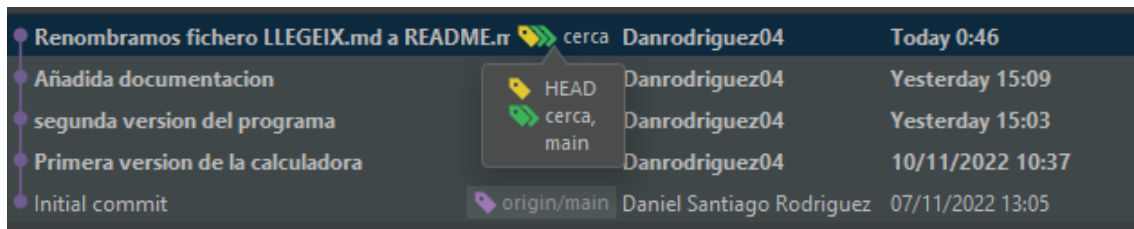




## -Ordenación

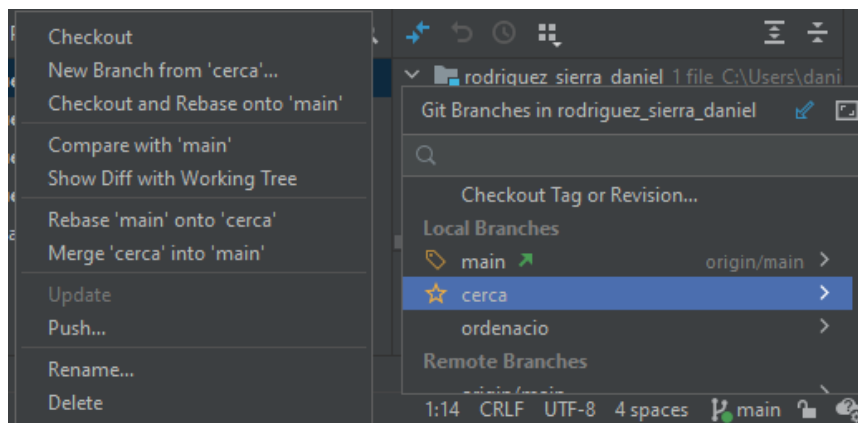
La calculadora ordena nombres	ordenacio	Danrodriguez04	47 minutes ago
Renombramos fichero LLEGEIX.md a README.r	cerca	Danrodriguez04	Today 0:46
Añadida documentacion		Danrodriguez04	Yesterday 15:09
segunda version del programa		Danrodriguez04	Yesterday 15:03
Primera version de la calculadora		Danrodriguez04	10/11/2022 10:37
Initial commit	origin/main	Daniel Santiago Rodriguez	07/11/2022 13:05

-Cerca – main

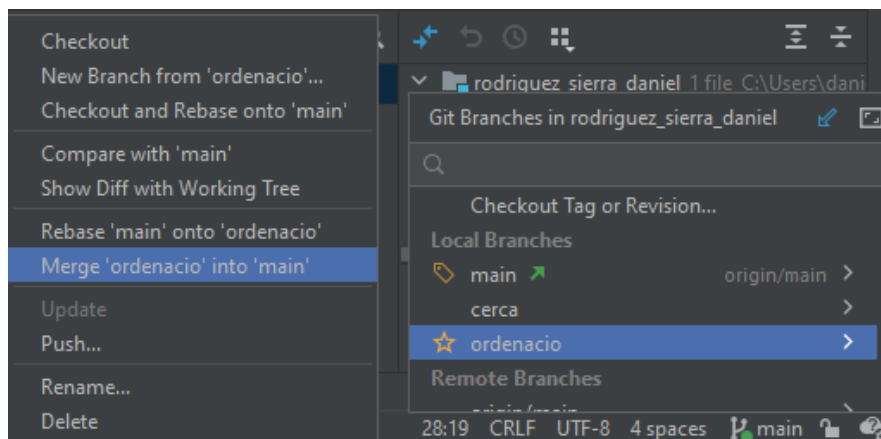


Si se podría fusionar el contenido de cerca y ordenacio dentro de main y el contenido que hay en ordenacio y cerca quedaran en main, para hacerlo se hará un git merge

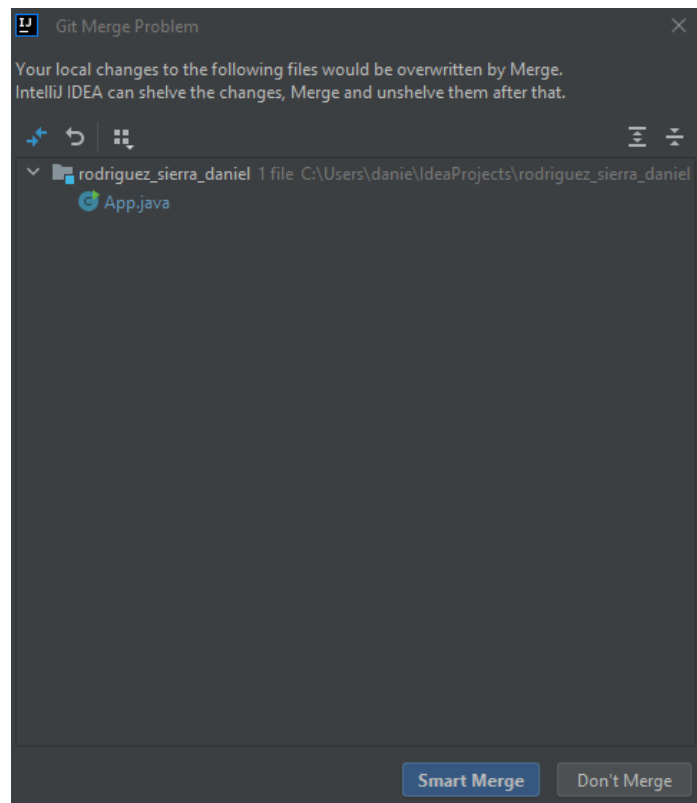
Git merge



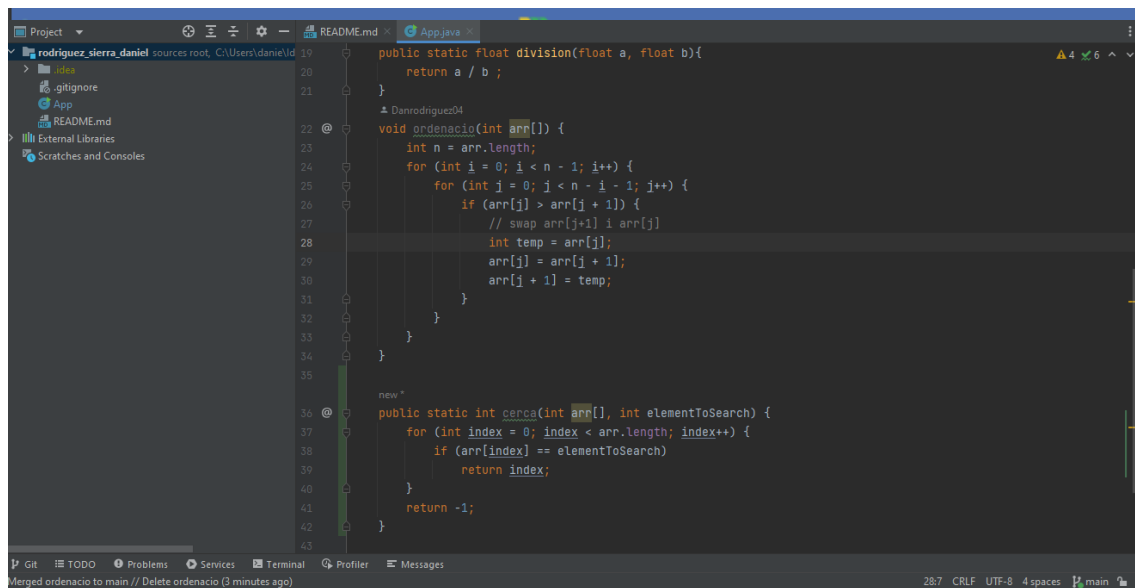
Al fusionar cerca y main he podido ver que como no se ha hecho ningún commit en cerca para que se guardara su versión, ha tenido el mismo contenido que main



Al fusionar ordenacio y main ya se han podido ver los cambios que se han hecho, ya que ha tenido una versión de su rama

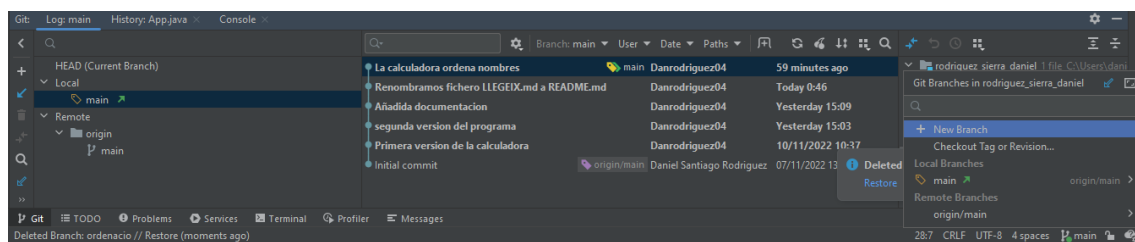
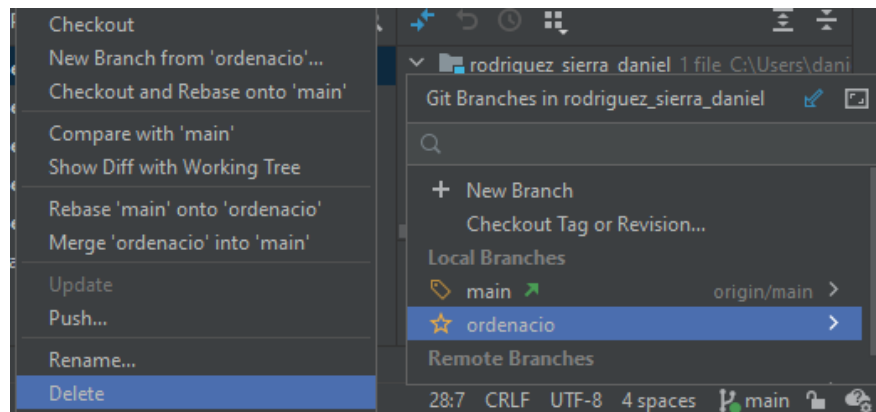


(Fusión de contenido de las ramas)



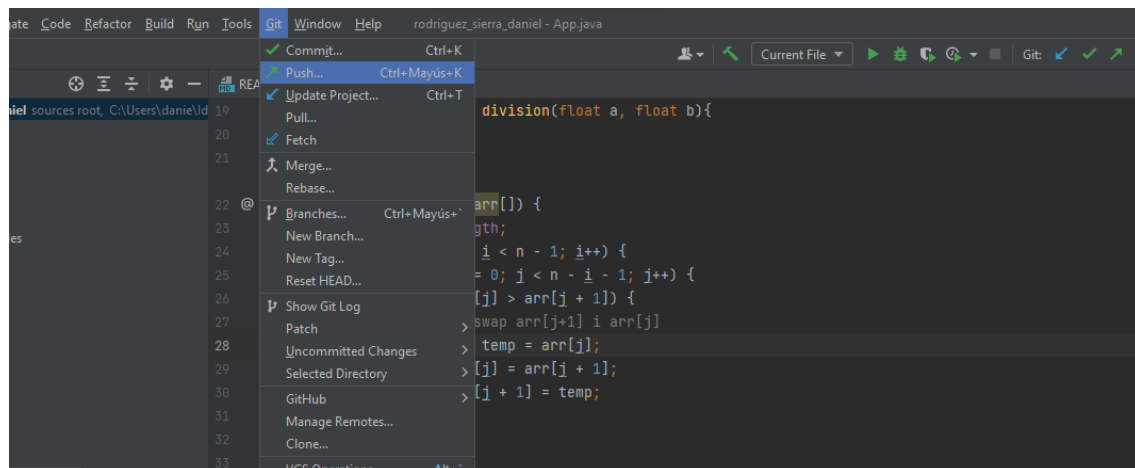
### R13. Borrarnos las ramas sobrantes

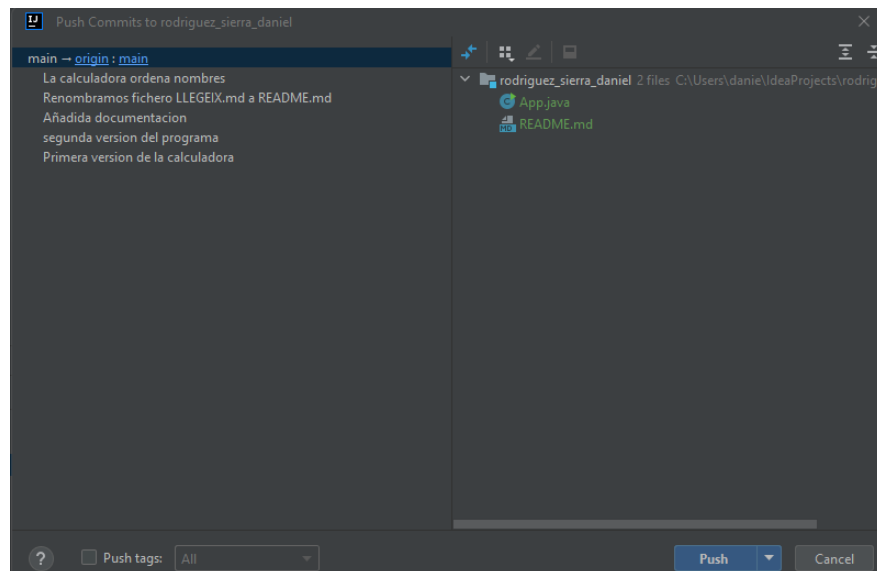
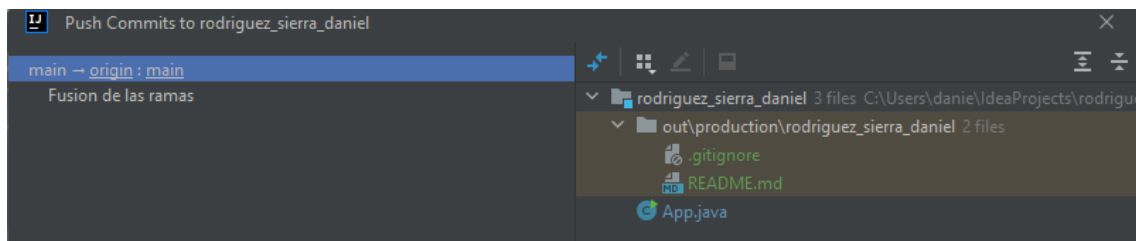
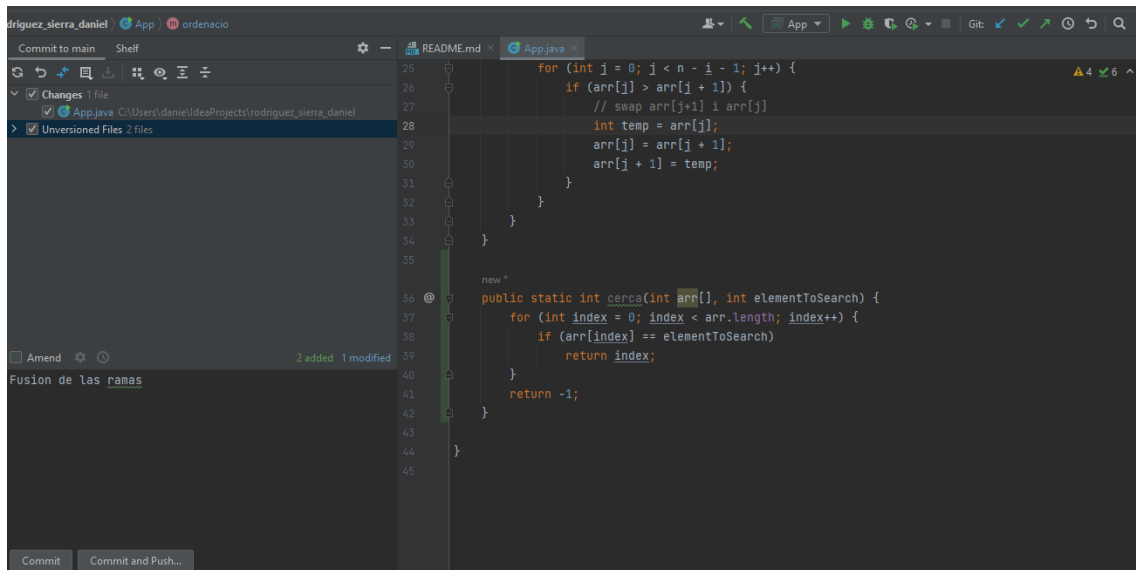
Git Branch -d “y el nombre de la rama que queremos borrar”



### R14. Subimos los cambios a Bitbucket

git add, git commit y git push

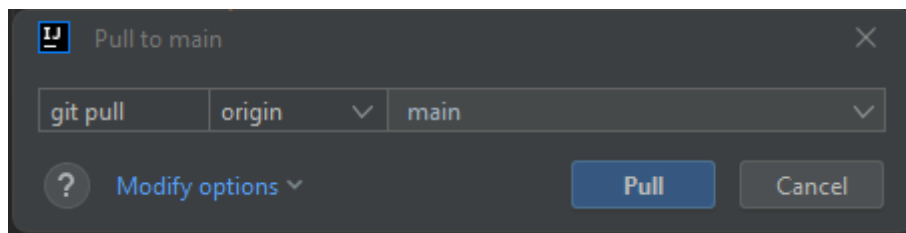
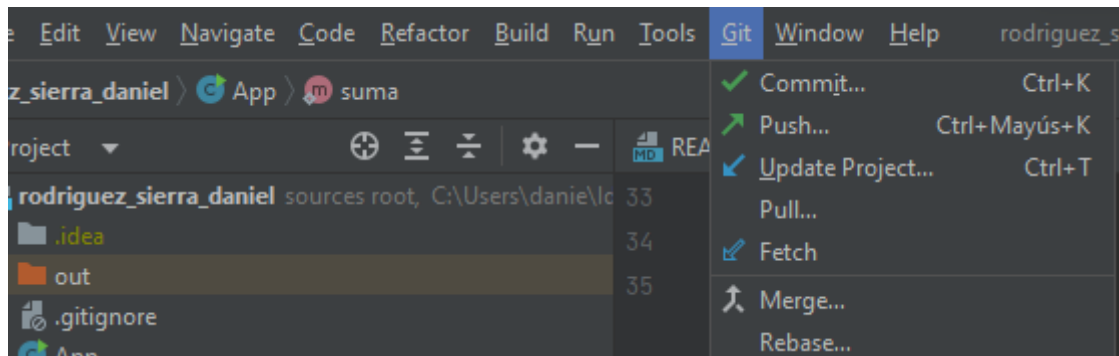




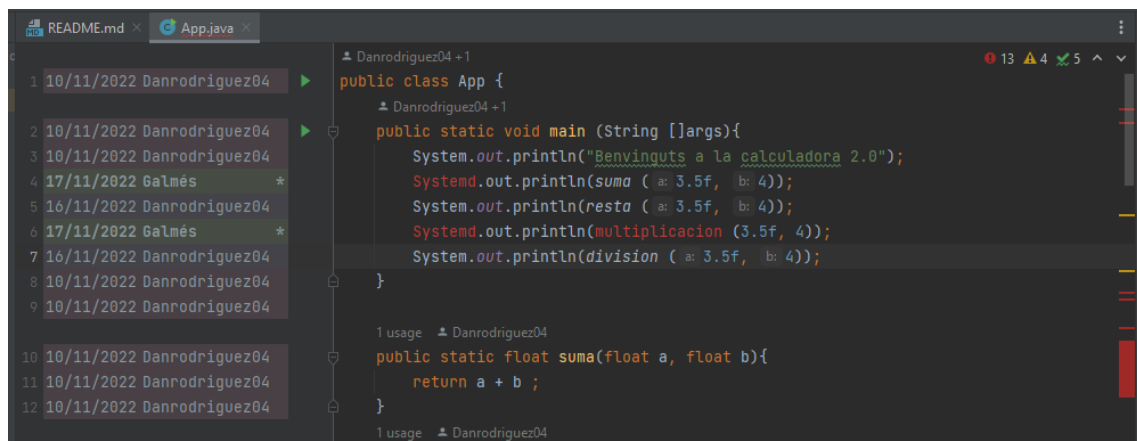


## R15. Resolvemos problemas

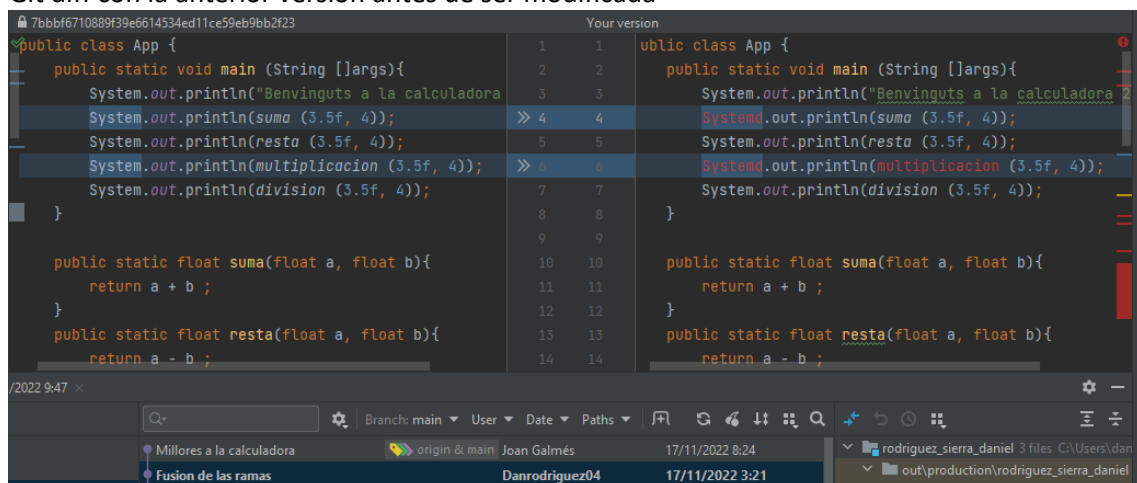
### Git pull



(Cambios hechos por el profesor )



Git diff con la anterior versión antes de ser modificada



```
7bbb6f710889f39e6614534ed11ce59eb9bb2f23      Your version
public static float division(float a, float b){    19 19 public static float division(float a, float b){
    return a / b ;                                20 20     return a / b ;
}                                                    21 21 }
                                                    22 22 @
void ordenacio(int arr[]) {                        23 23 void ordenar(int arr[]) {
    int n = arr.length;                            24 24     int n = arr.length;
    for (int i = 0; i < n - 1; i++) {               25 25         for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {        26 26             arr[j] = arr[j + 1];
            if (arr[j] > arr[j + 1]) {                27 27             arr[j + 1] = temp;
                // swap arr[j+1] i arr[j]              28 28         }
                int temp = arr[j];                    29 29     }
                arr[j] = arr[j + 1];                  30 30 }
                arr[j + 1] = temp;                    31 31 }
            }                                           32 32 public static int cerca(int arr[], int elementToS
        }
    }
}
```

Para corregir el código debo volver a crear las variables que se han eliminado

## R16. ¿Quién ha hecho que?

El código no compila debido a que se han eliminado algunas variables que necesitan ser declaradas para que el programa pueda compilar el código