



# P3 T4

## GIT HUB / GIT

**MARIO JIMÉNEZ MARSET**

**ÍNDICE**

1. ENUNCIADO - OBJETIVOS.....	3
2. DESARROLLO – PROCEDIMIENTOS.....	3

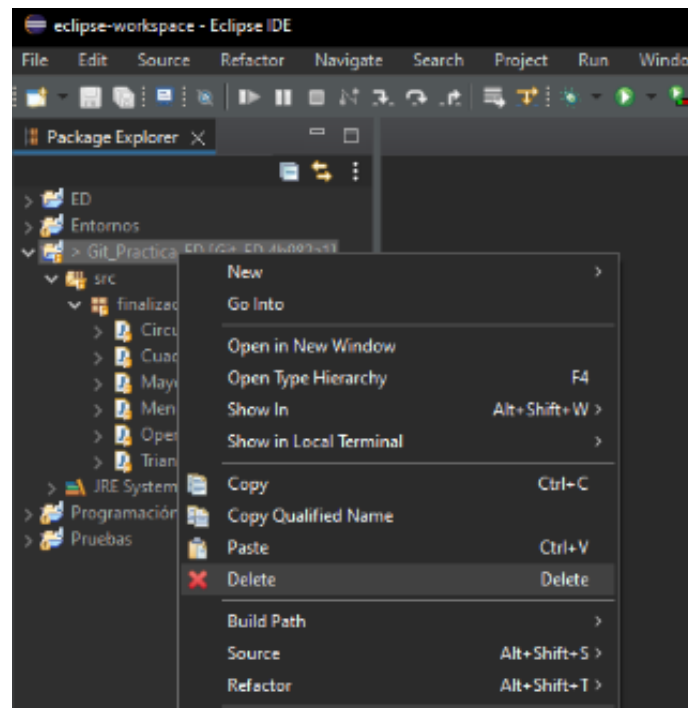
## 1. ENUNCIADO - OBJETIVOS

En esta práctica se pedía realizar varios pasos de control de versiones dentro de GitHub y Eclipse, los cuáles eran:

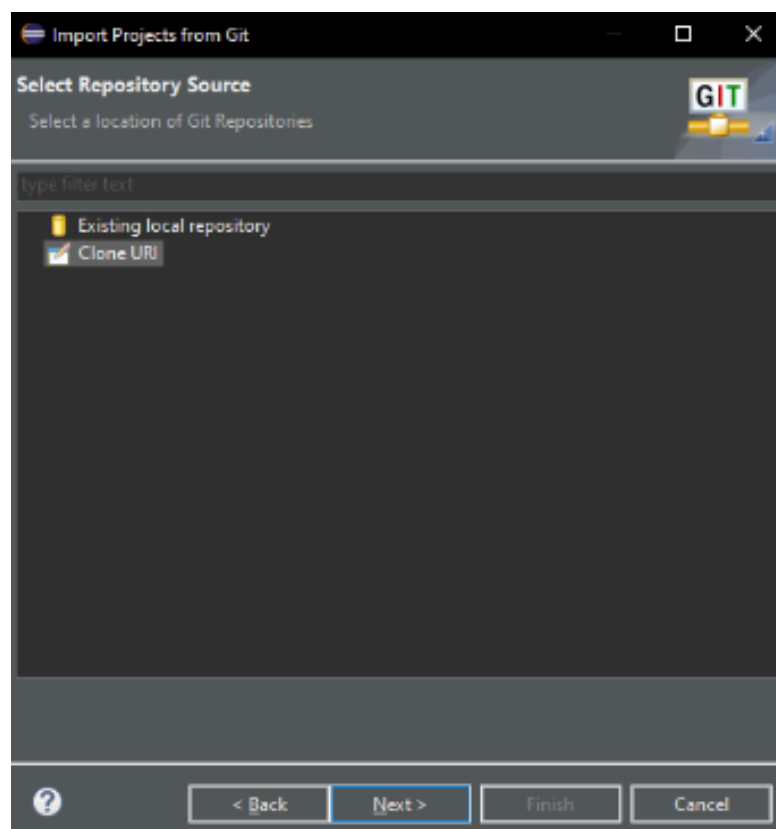
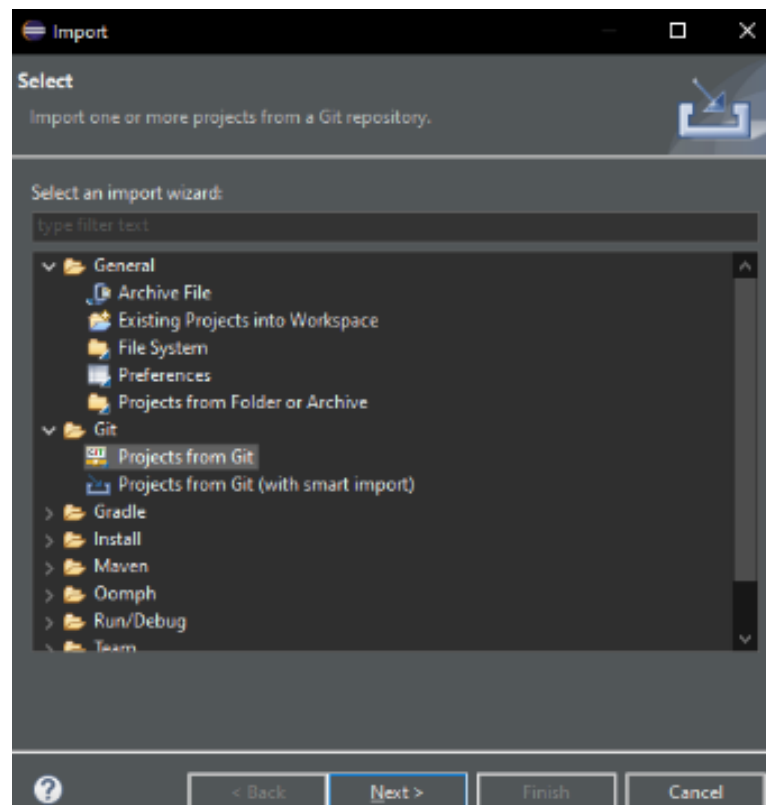
- 1) En Eclipse, borrar el proyecto que previamente habíais subido a GitHub (con Push) en la Práctica 2. Mediante un clone, importar el proyecto desde GitHub a Eclipse.
- 2) En GitHub, en el repositorio remoto, modificar algún archivo del proyecto y realizar el commit.
- 3) En Eclipse, actualizar el repositorio local (un Pull) con el cambio realizado en GitHub.
- 4) En GitHub, crear una nueva rama con algún cambio en fichero (o añadido otro fichero) de la rama master. Hacer commit en dicha rama.
- 5) En GitHub hacer un merge de ambas ramas (pero sin eliminar ninguna).
- 6) En Eclipse, actualizar en el repositorio local, la nueva rama del repositorio remoto.
- 7) En Eclipse, cambiar (checkout) a la nueva rama, hacer un cambio y subirlo a Github. Comprobar en GitHub que se ha subido correctamente.

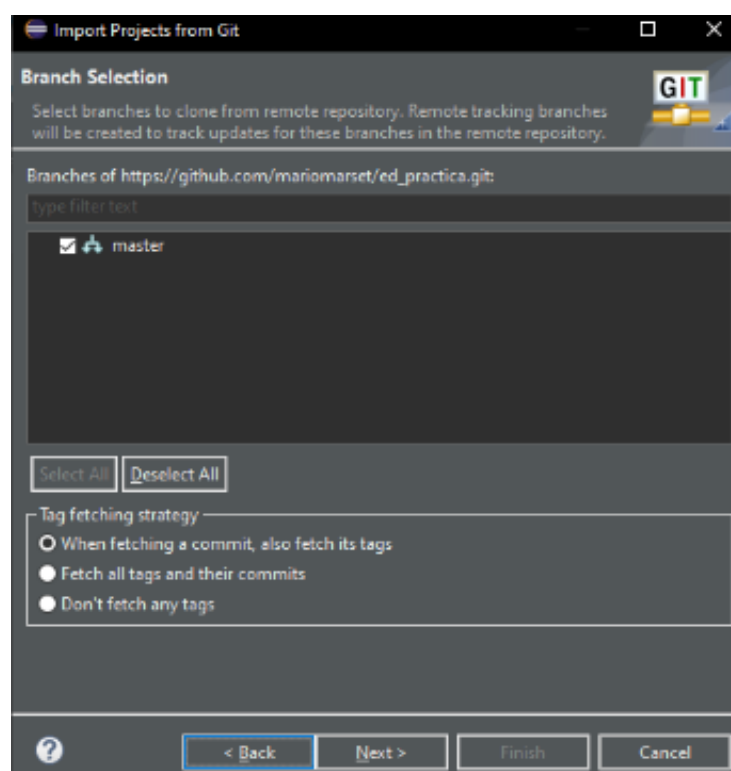
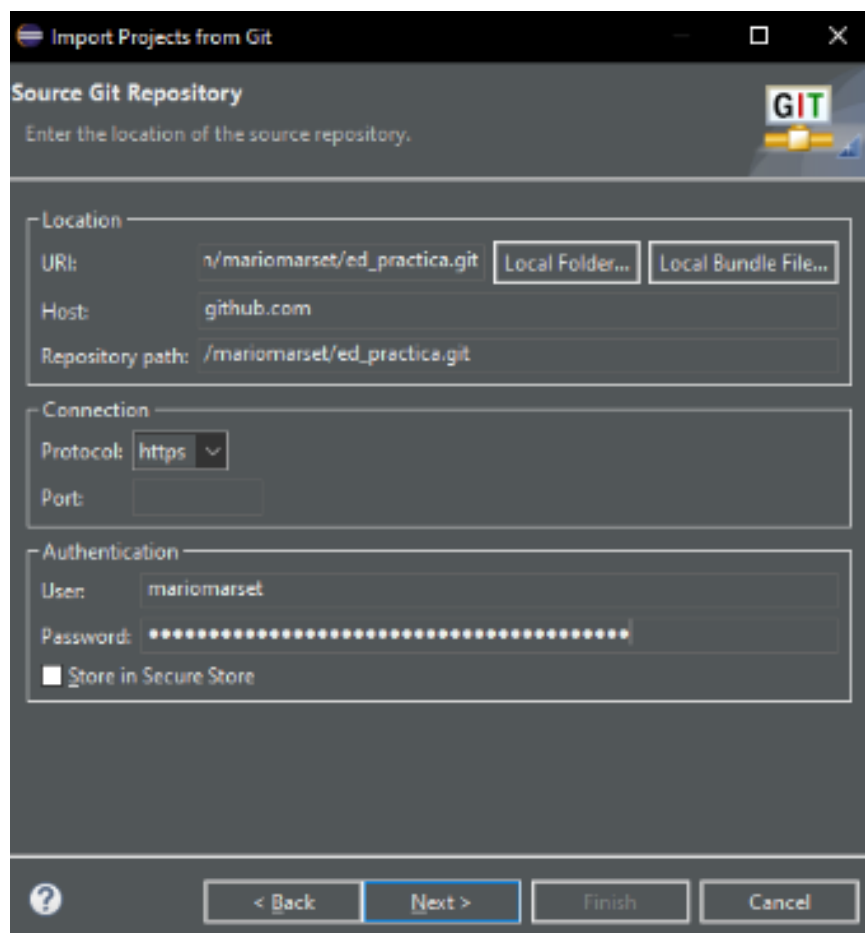
## 2. DESARROLLO – PROCEDIMIENTOS

Se adjuntan las capturas de pantalla y las explicaciones necesarias para el entendimiento de las mismas.

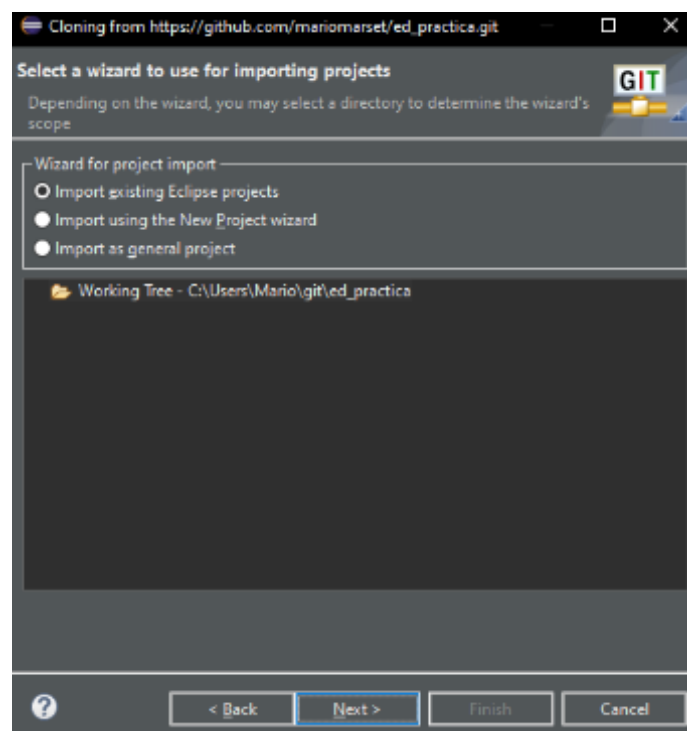
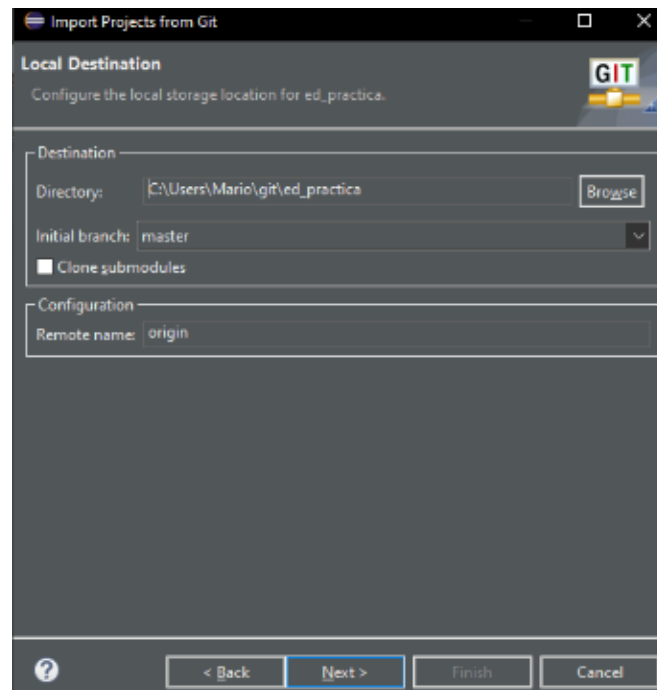


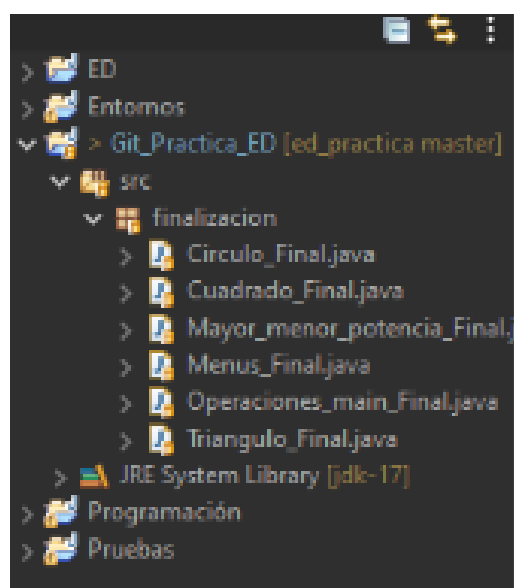
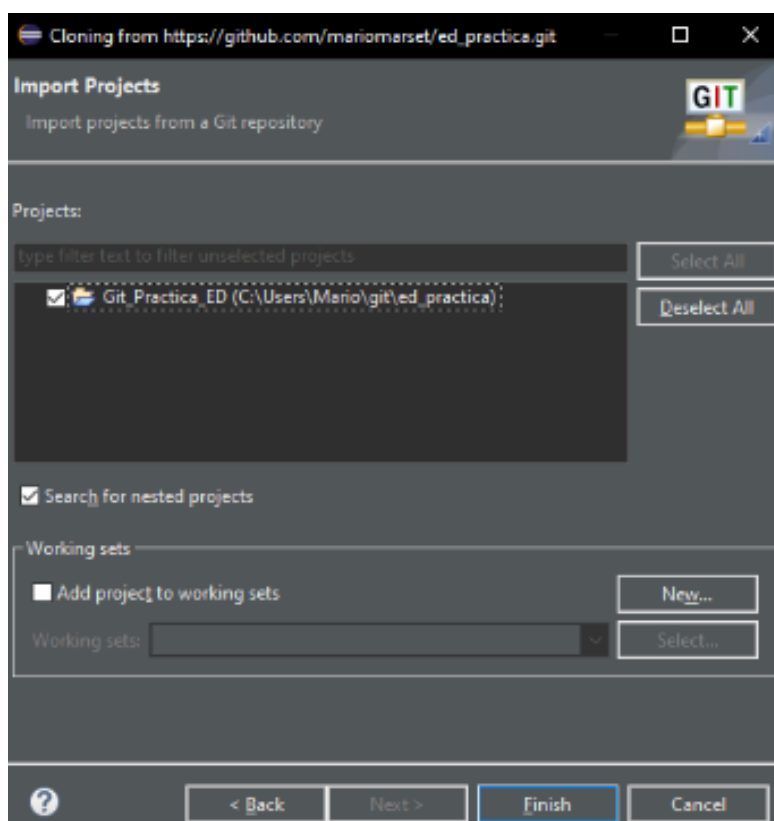
Se borra el proyecto dentro de Eclipse.



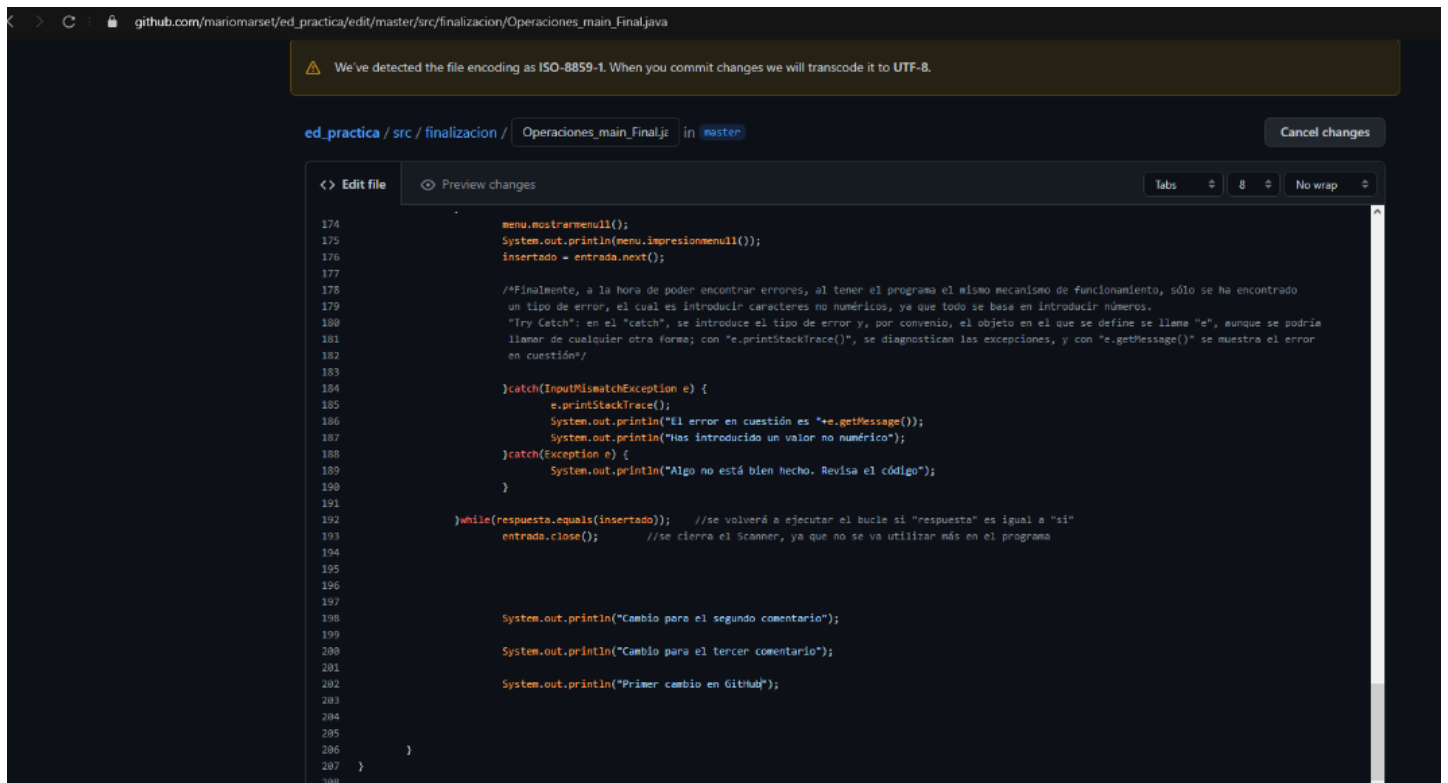


En estas capturas, se importa el proyecto desde File → Import → Projects from Git → Clone URI → URI (User y Password de GitHub) → Master.



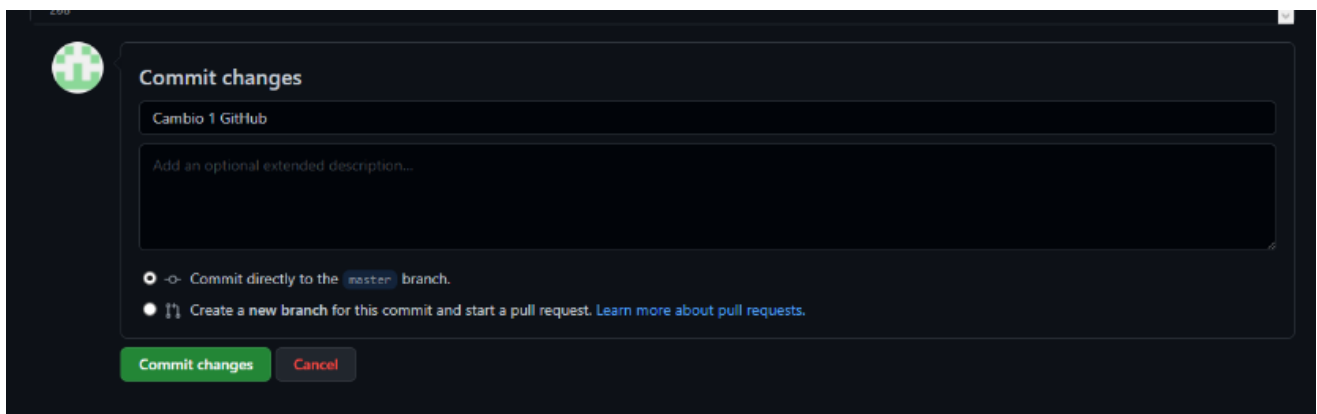


Se muestra que se ha importado el proyecto desde GitHub hasta Eclipse.



```
174         menu.mostrarmenu1();
175         System.out.println(menu.impressionmenu1());
176         insertado = entrada.next();
177
178         /*Finalmente, a la hora de poder encontrar errores, al tener el programa el mismo mecanismo de funcionamiento, sólo se ha encontrado
179         un tipo de error, el cual es introducir caracteres no numéricos, ya que todo se basa en introducir números.
180         "Try Catch": en el "catch", se introduce el tipo de error y, por convenio, el objeto en el que se define se llama "e", aunque se podría
181         llamar de cualquier otra forma; con "e.printStackTrace()", se diagnostican las excepciones, y con "e.getMessage()" se muestra el error
182         en cuestión*/
183
184     }catch(InputMismatchException e) {
185         e.printStackTrace();
186         System.out.println("El error en cuestión es "+e.getMessage());
187         System.out.println("Has introducido un valor no numérico");
188     }catch(Exception e) {
189         System.out.println("Algo no está bien hecho. Revisa el código");
190     }
191
192     }while(respuesta.equals(insertado)); //se volverá a ejecutar el bucle si "respuesta" es igual a "si"
193     entrada.close(); //se cierra el Scanner, ya que no se va utilizar más en el programa
194
195
196
197
198     System.out.println("Cambio para el segundo comentario");
199
200     System.out.println("Cambio para el tercer comentario");
201
202     System.out.println("Primer cambio en GitHub");
203
204
205
206     }
207 }
```

Dentro de GitHub se realiza una impresión por pantalla (lo cual modifica el archivo "Operaciones\_main\_Final.java" del proyecto) el cual dice "Primer cambio en GitHub".



**Commit changes**

Cambio 1 GitHub

Add an optional extended description...

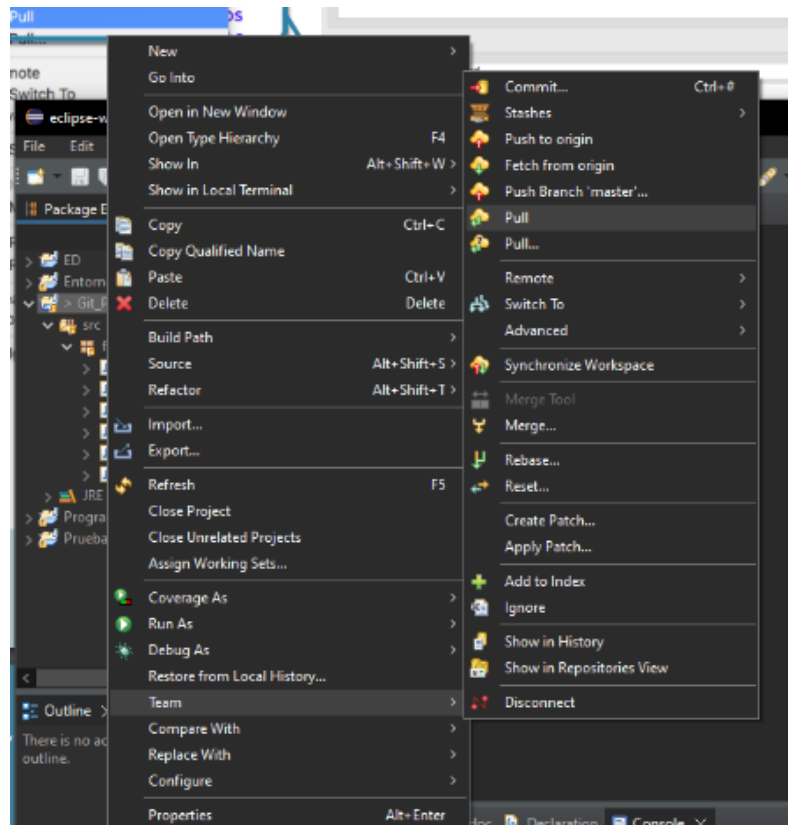
☒ Commit directly to the `master` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

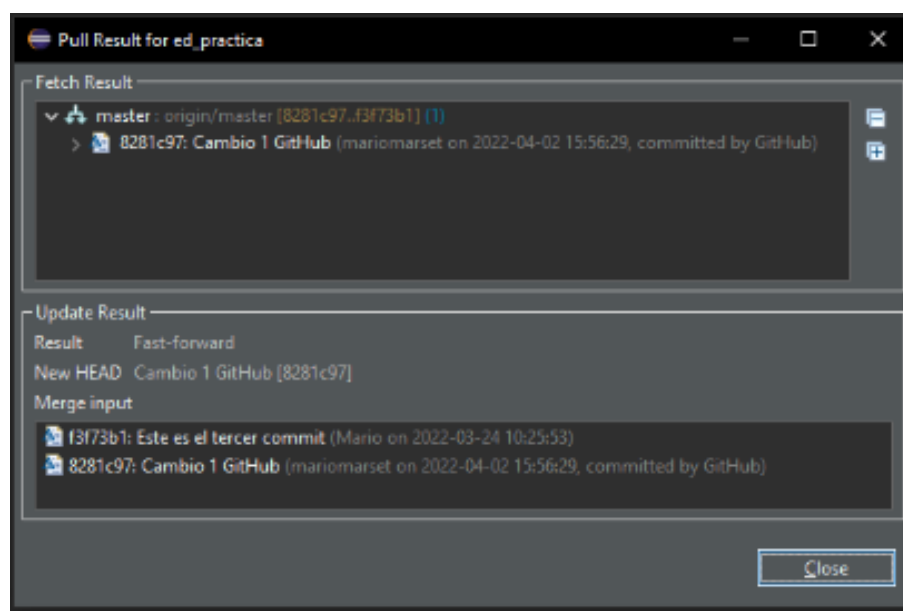
**Commit changes** **Cancel**

Para realizar el commit, se le incluye un título.



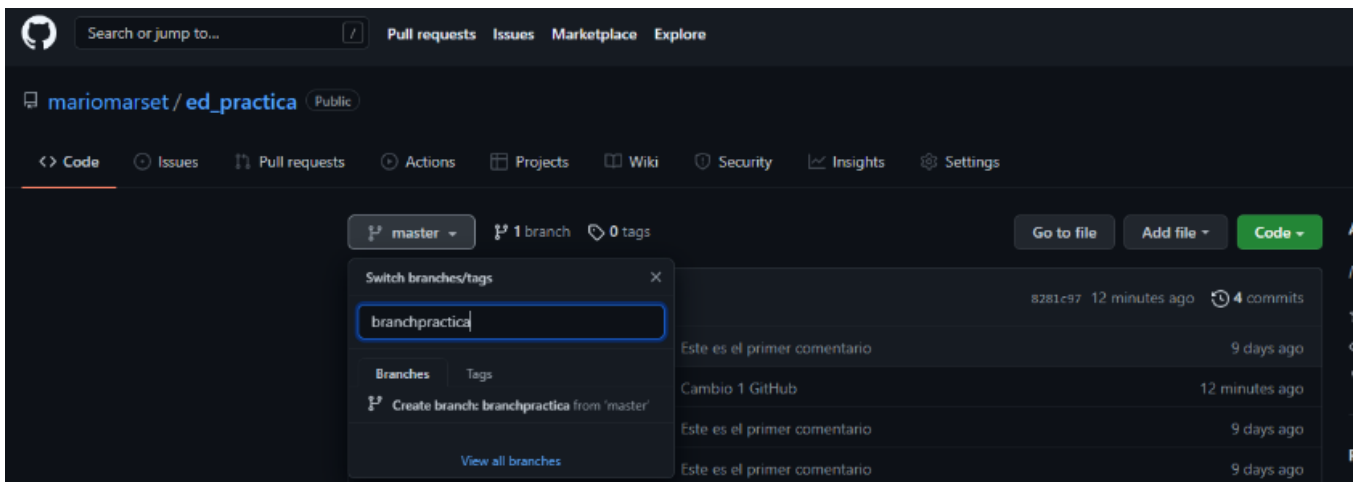


Para actualizar el repositorio local dentro de Eclipse, se clic en el Proyecto → Team → Pull.

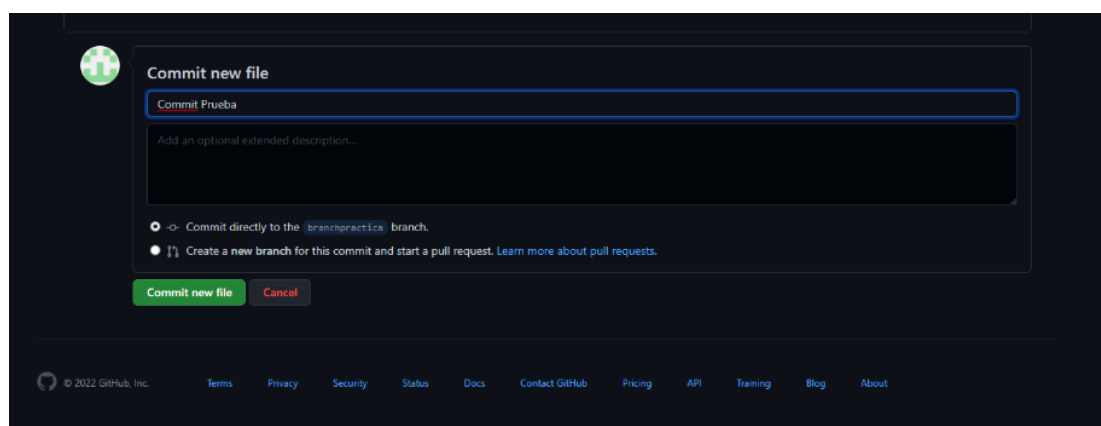
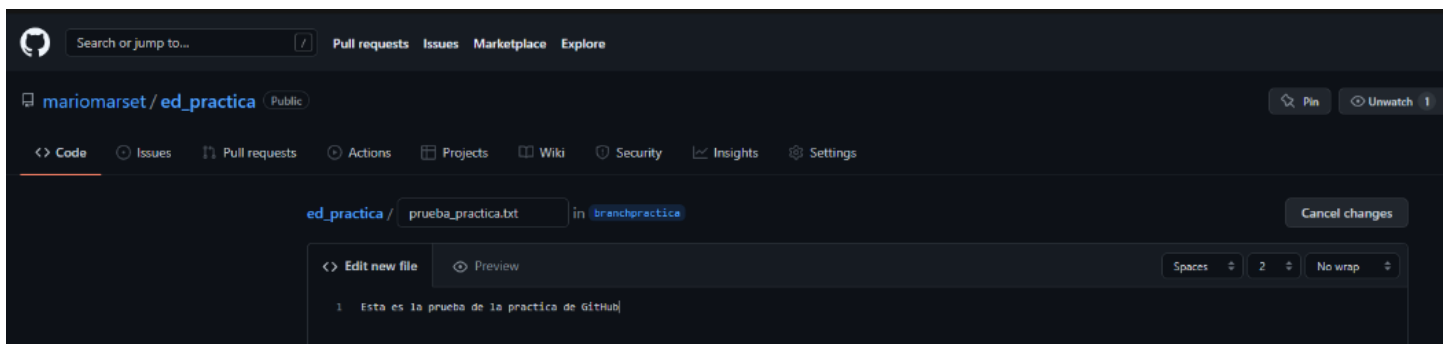


Se muestra el commit hecho dentro de GitHub.

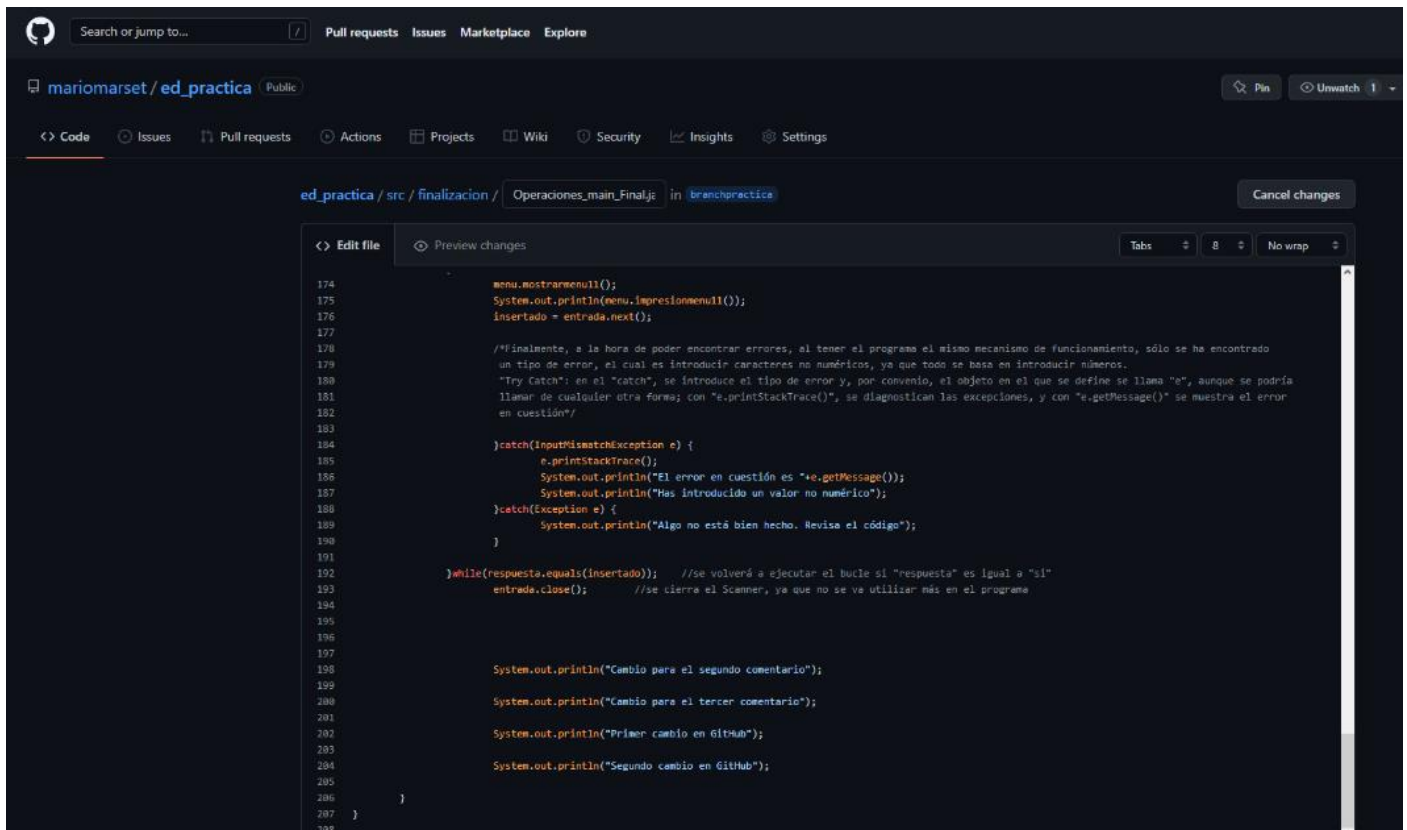
Dentro de GitHub, se crea un branch o rama, de nombre “branchpractica”.



Hecho esto, se crea un nuevo fichero.

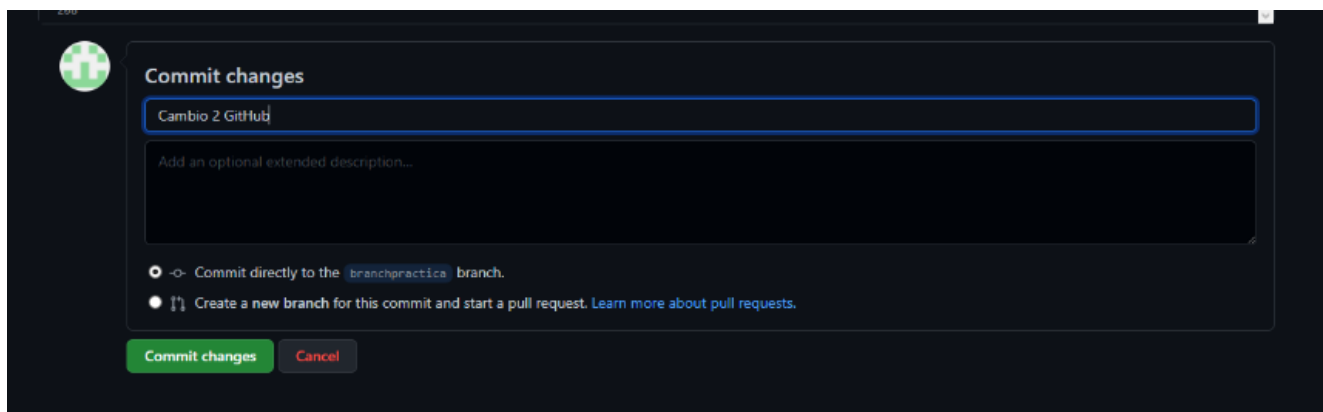


Se debe hacer siempre commit.



```
174 menu.mostrarMenu();
175 System.out.println(menu.getMenu());
176 insertado = entrada.next();
177
178 /*Finalmente, a la hora de poder encontrar errores, el tener el programa el mismo mecanismo de funcionamiento, sólo se ha encontrado
179 un tipo de error, el cual es introducir caracteres no numéricos, ya que todo se basa en introducir números.
180 "Try Catch": en el "catch", se introduce el tipo de error y, por convenio, el objeto en el que se define se llama "e", aunque se podría
181 llamar de cualquier otra forma; con "e.printStackTrace()", se diagnostican las excepciones, y con "e.getMessage()" se muestra el error
182 en cuestión*/
183
184 }catch(InputMismatchException e) {
185     e.printStackTrace();
186     System.out.println("El error en cuestión es "+e.getMessage());
187     System.out.println("Has introducido un valor no numérico");
188 }catch(Exception e) {
189     System.out.println("Algo no está bien hecho. Revisa el código");
190 }
191
192 }while(respuesta.equals(insertado)); //se volverá a ejecutar el bucle si "respuesta" es igual a "si"
193 entrada.close(); //se cierra el Scanner, ya que no se va utilizar más en el programa
194
195
196
197
198 System.out.println("Cambio para el segundo comentario");
199
200 System.out.println("Cambio para el tercer comentario");
201
202 System.out.println("Primer cambio en GitHub");
203
204 System.out.println("Segundo cambio en GitHub");
205
206
207 }
208
```

Se realiza un cambio en un archivo dentro del proyecto (dentro del branch nuevo creado) y se realiza el commit (obligatorio siempre).



**Commit changes**

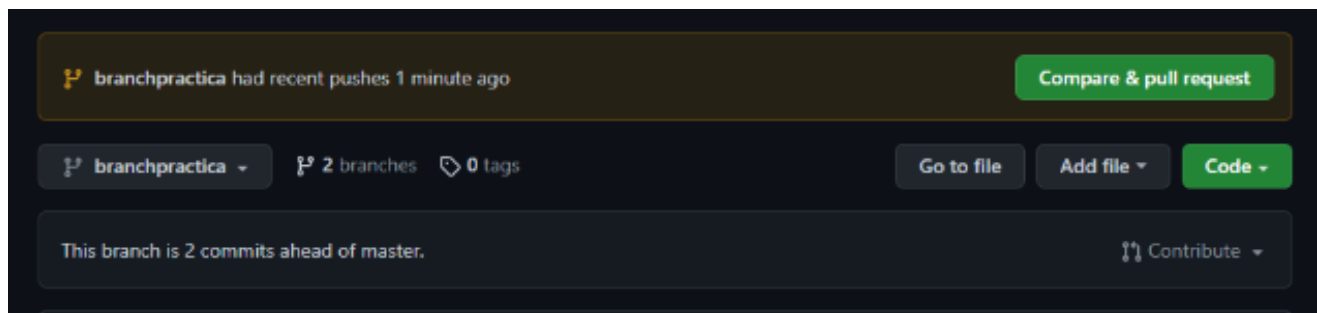
Cambio 2 GitHub

Add an optional extended description...

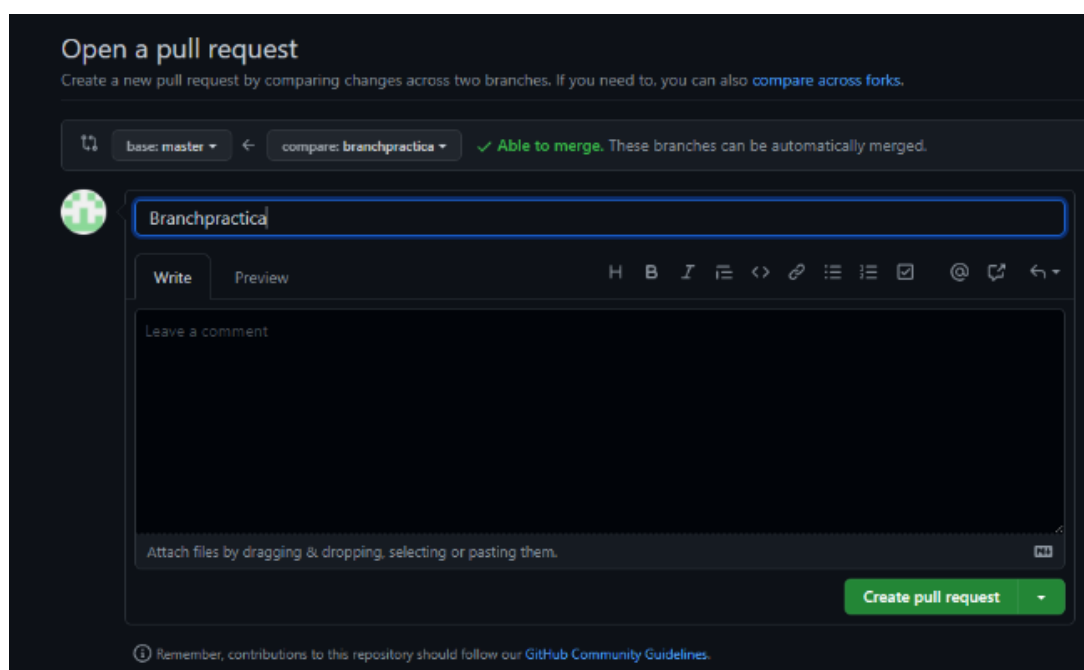
☒ Commit directly to the `branchpractice` branch.

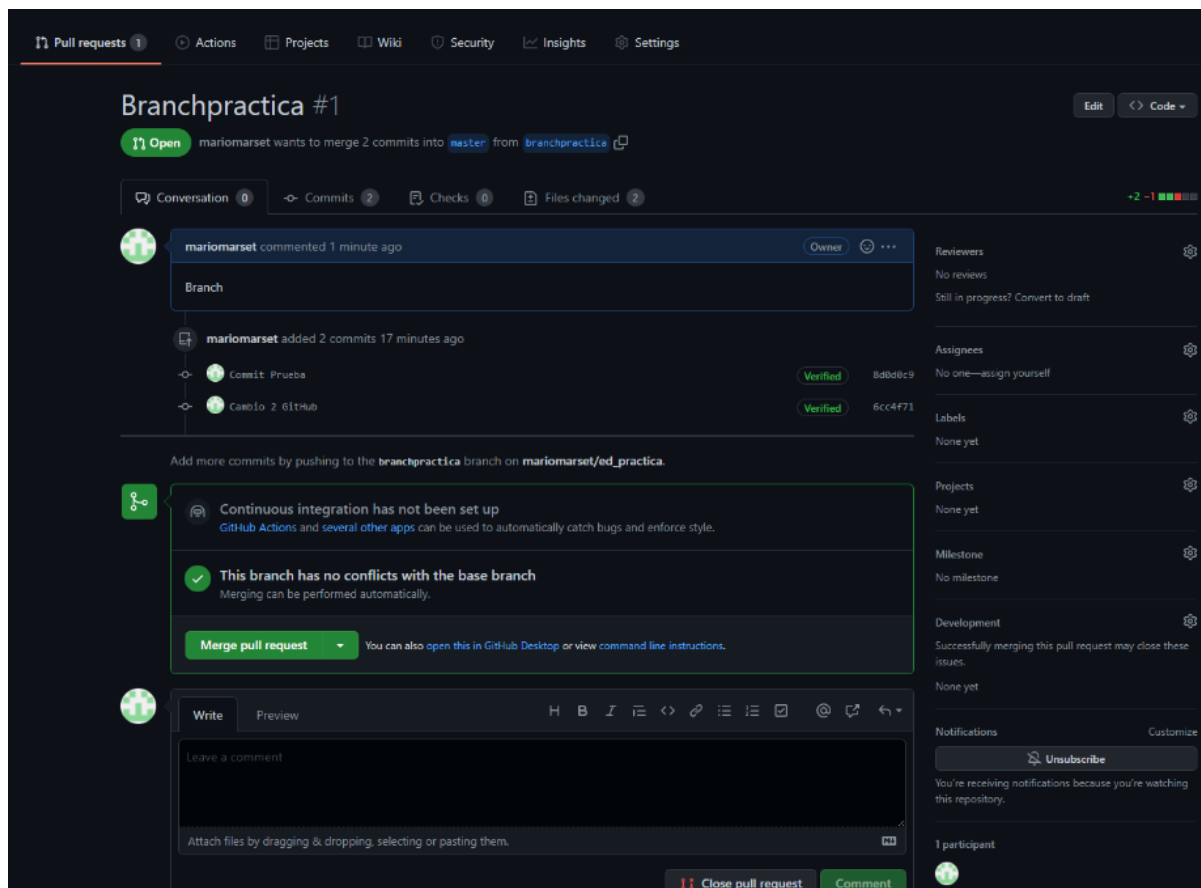
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

**Commit changes** **Cancel**

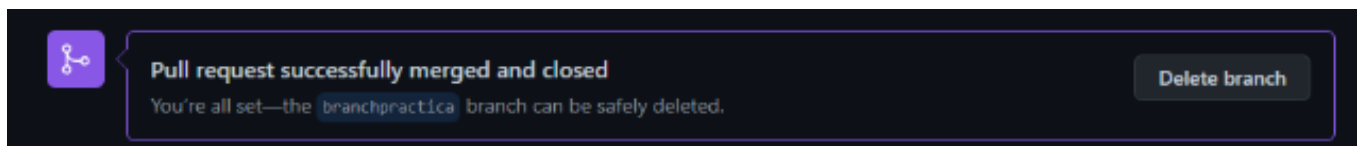
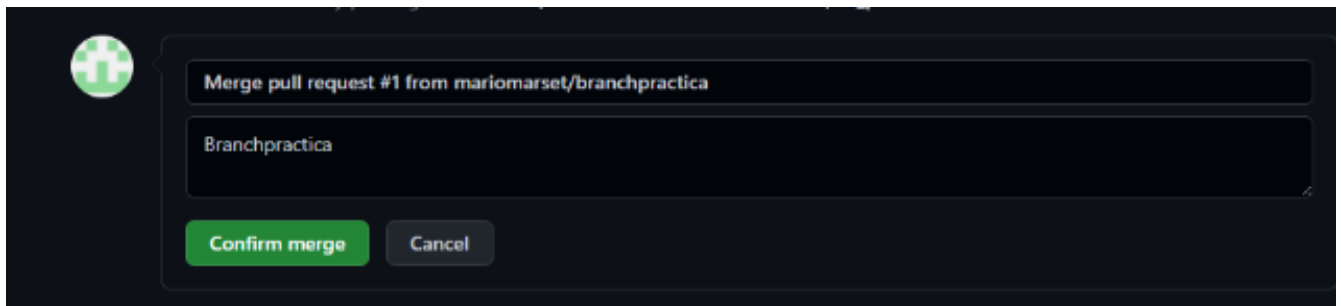


Dentro de GitHub también, se crea un pull request.

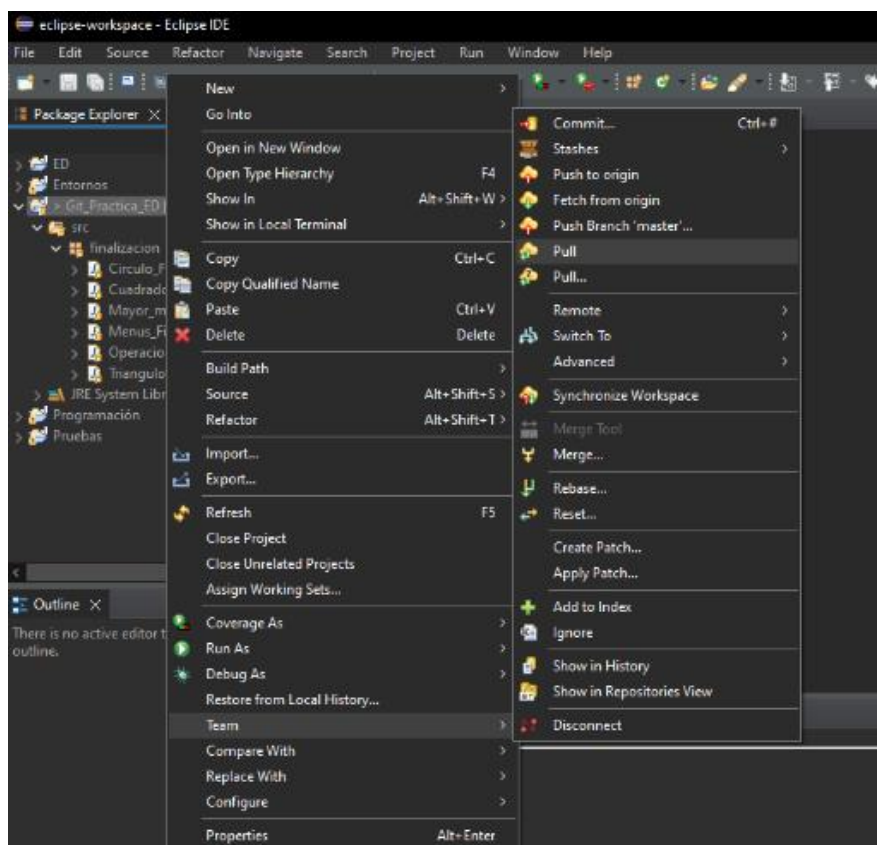




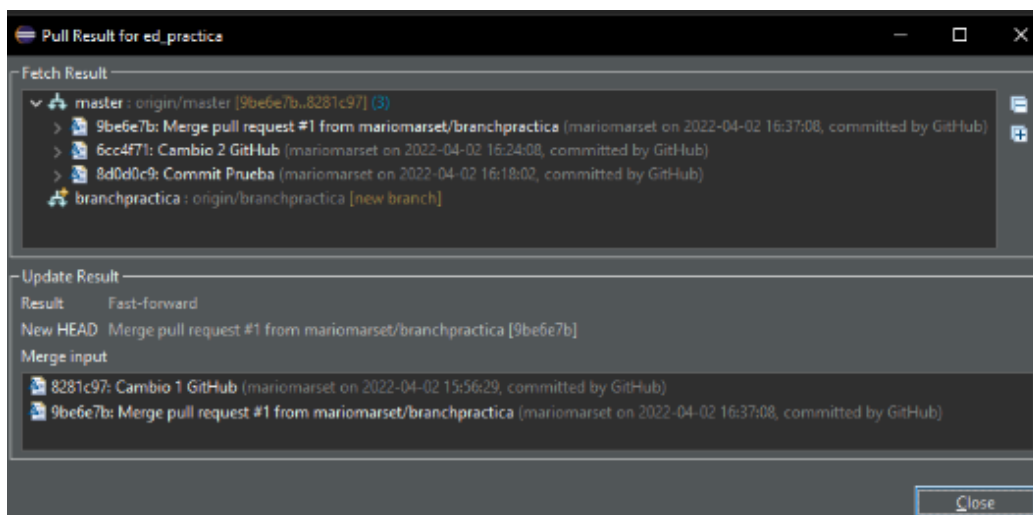
Se hace el “merge pull request” y se confirma el merge.

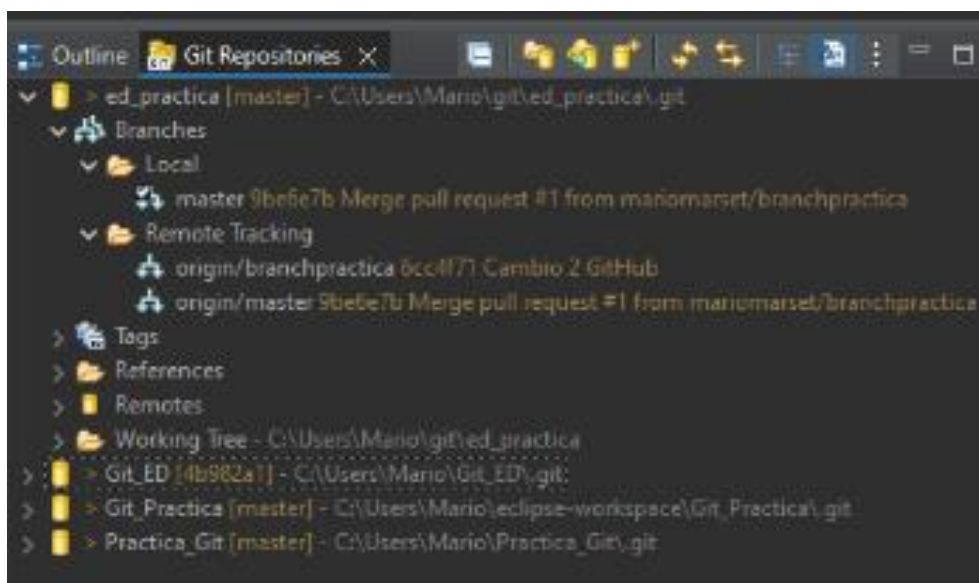


Dentro de Eclipse, se actualiza en el repositorio local la nueva rama creada en GitHub.



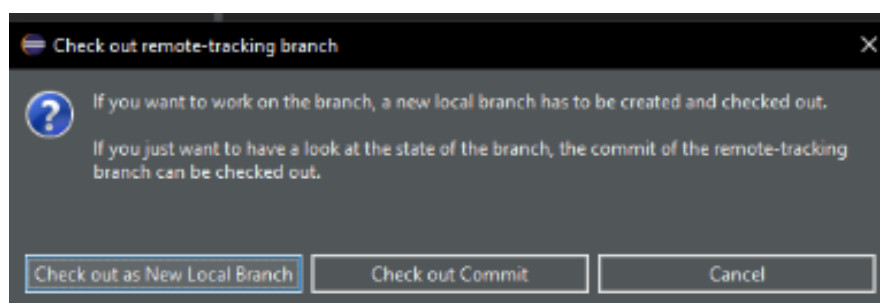
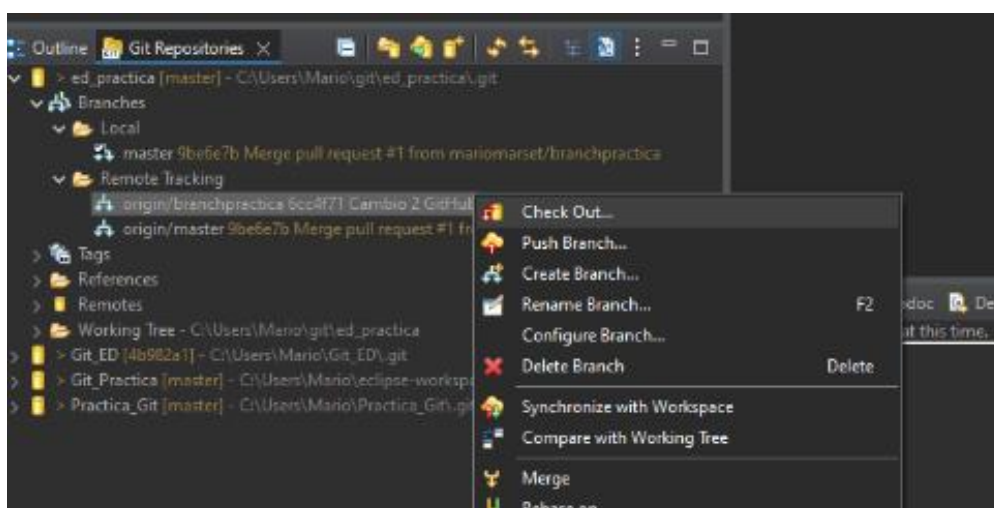
Al igual que en anteriores pasos, se clic en el Proyecto → Team → Pull.

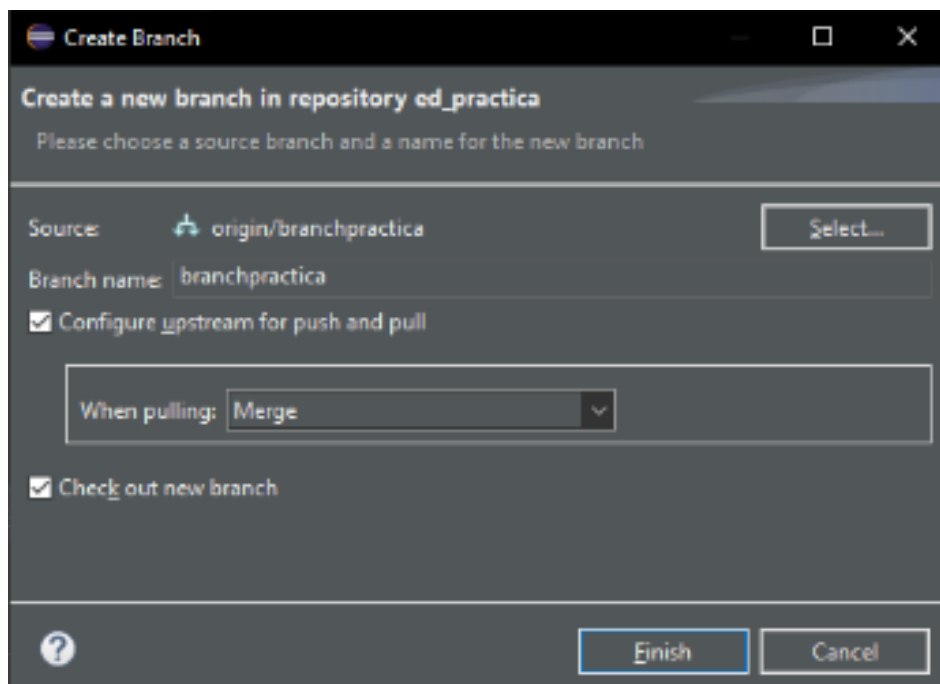




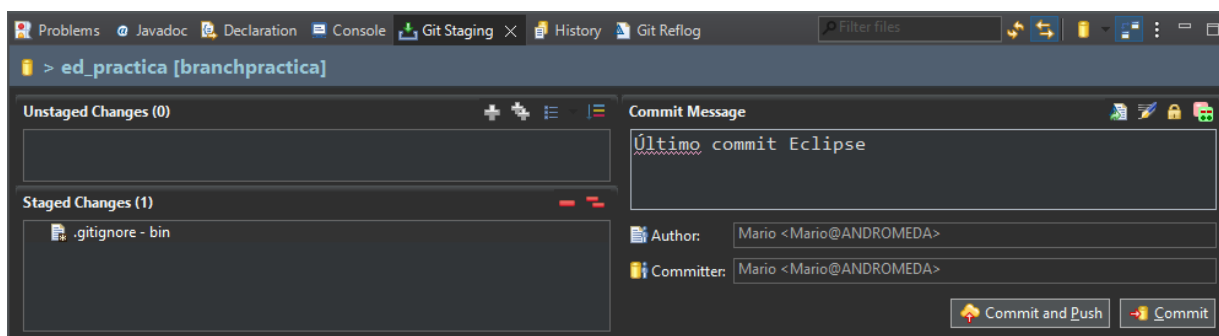
Con la vista Git Repositories se comprueban las distintas ramas en los 2 repositorios.

Se hace un checkout a la segunda rama para traerla al repositorio local en Eclipse.



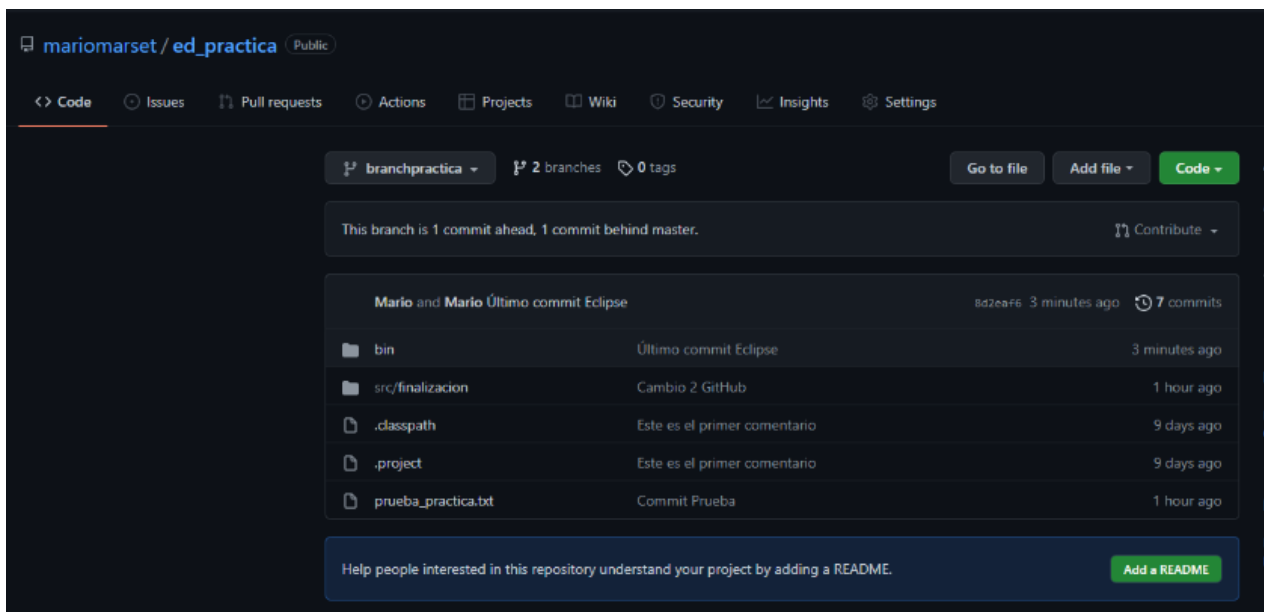


Al finalizar, se visualiza que se mantiene el nombre de la nueva rama que había en GitHub.



Se realiza el último cambio en Eclipse y se sube con commit and push a GitHub.





Se visualiza dentro de GitHub (dentro de la rama) que se ha hecho un cambio: se ha introducido una impresión por pantalla en uno de los archivos del proyecto: “Último commit Eclipse”.