

Proyecto de Práctica con Git y GitHub – Java + Eclipse + GitHub Desktop

1. Datos del Proyecto

Nombre del proyecto: ProyectoGitJava

Nombre: David Pino

Fecha: 07-10-2025

2. Descripción General

El objetivo de este proyecto es practicar el uso de **Git y GitHub** mediante la creación de un proyecto Java con varias ramas, commits y merges, aplicando los conceptos de control de versiones vistos en clase.

Durante el desarrollo se han realizado merges con y sin conflictos, y se ha documentado cada paso del proceso.

3. Estructura de Ramas

El repositorio cuenta con la siguiente estructura de ramas:

Rama	Descripción
main	Rama principal del proyecto. Contiene el código final y estable.
saludo	Rama secundaria donde se añadió una función de saludo.
despedida	Rama secundaria donde se añadió una función de despedida.

4. Commits Realizados

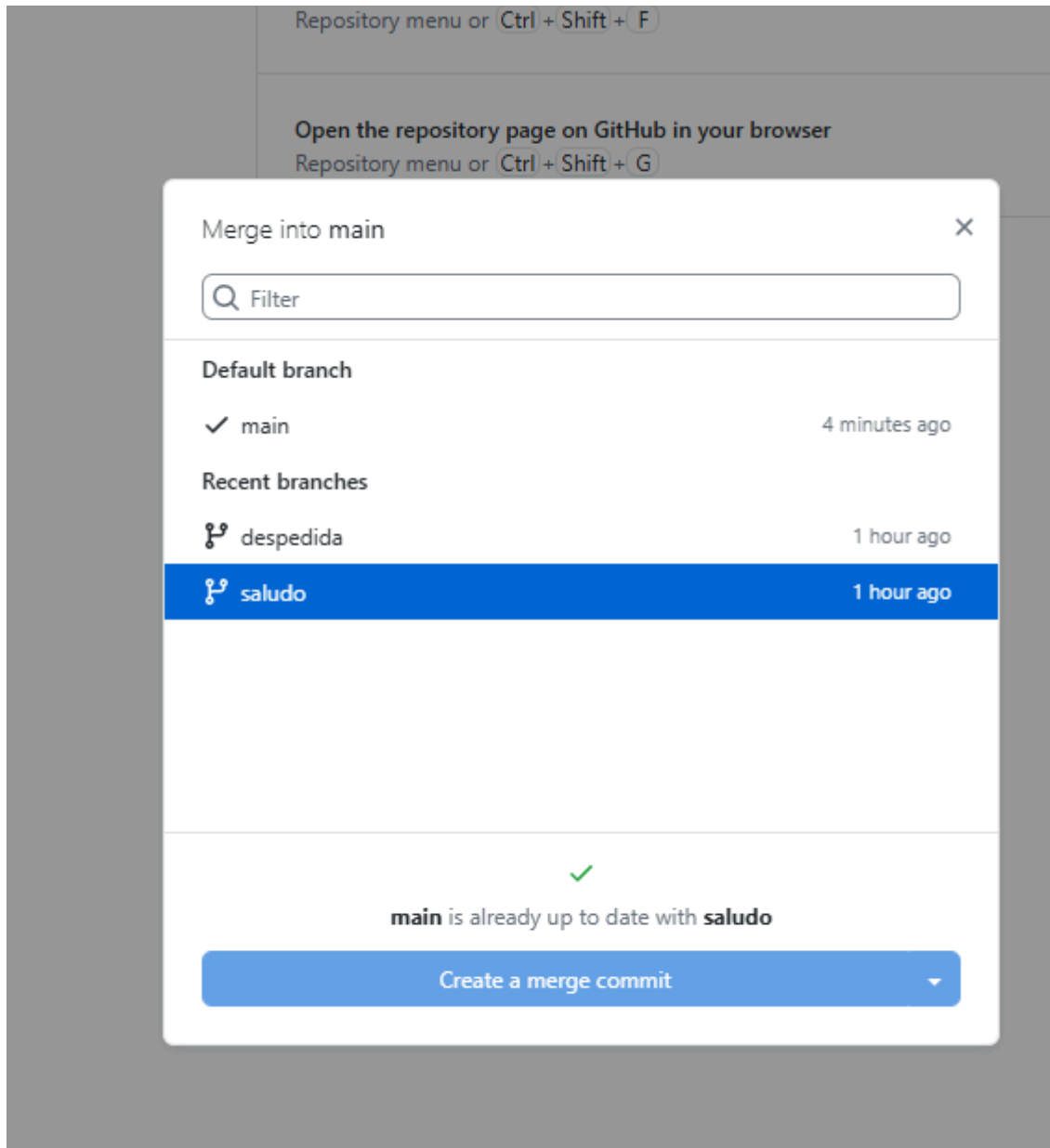
Rama	Commit	Descripción
------	--------	-------------

main	Versión inicial del proyecto Java	Se crea el proyecto base con un <code>System.out.println()</code> inicial.
saludo	Añadida función saludar	Se crea una función nueva llamada <code>saludar</code> .
saludo	Mejora del mensaje de saludo	Se modifica el texto del mensaje.
despedida	Añadida función <code>despedirse()</code>	Se implementa la función <code>despedirse</code> .
despedida	Corrección de formato en la despedida	Se mejora el texto mostrado por consola.

5. Merge sin Conflicto

- **Rama fusionada:** `saludo` → `main`

- **Resultado:** Merge realizado correctamente sin errores.



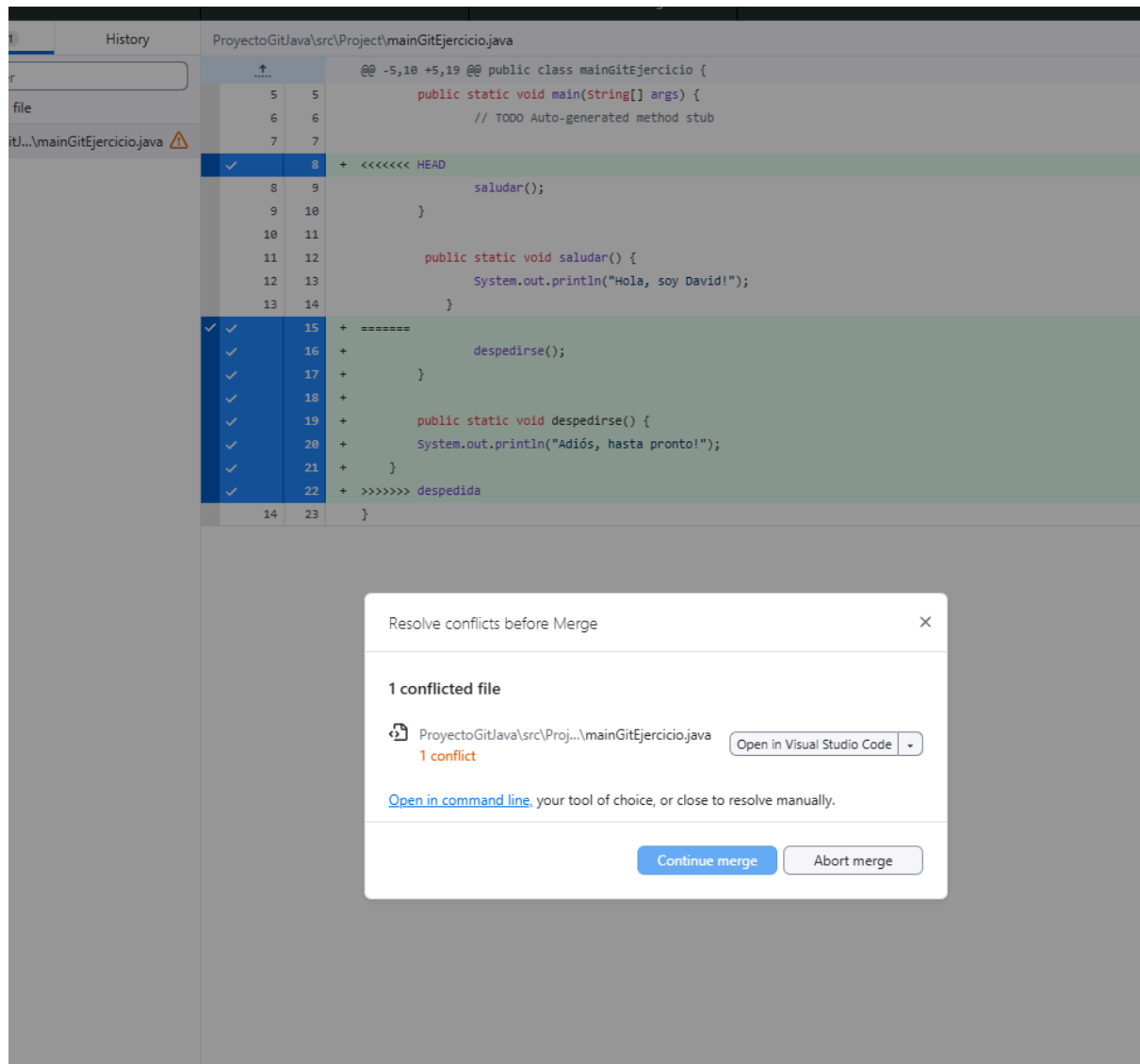
6. Merge con Conflicto y Resolución

- **Rama fusionada:** despedida → main
- **Resultado:** Se detectó un conflicto en el archivo Main.java.

Causa del conflicto

Ambas ramas modificaban la misma línea dentro del método `main()`, lo que provocó que Git no pudiera decidir qué versión conservar.

Código con conflicto



```
mainGitEjercicio.java X
E: > davidpino > mauri > ProyectoGitJavaPrueba > ProyectoGitJava > src > Project > mainGitEjercicio.java
1 package Project;
2
3 public class mainGitEjercicio {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
9         <<<<<< HEAD (Current Change)
10            saludar();
11        }
12
13        public static void saludar() {
14            System.out.println("Hola, soy David!");
15        }
16
17        =====
18        despedirse();
19    }
20
21    public static void despedirse() {
22        System.out.println("Adios, hasta pronto!");
23    }
24    >>>>>> despedida (Incoming Change)
25 }
```

Código final después de resolver el conflicto

```
File Edit Selection View Go ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

J mainGitEjercicio.java X
E: > davidpino > mauri > ProyectoGitJavaPrueba > ProyectoGitJava > src > Project > J mainGitEjercicio.java

1 package Project;
2
3 public class mainGitEjercicio {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         saludar();
9     }
10
11     public static void saludar() {
12         System.out.println("Hola, soy David!");
13     }
14     despedirse();
15 }
16
17 public static void despedirse() {
18     System.out.println("Adiós, hasta pronto!");
19 }
20 }
21
```

Resultado

```
System.out.println("Adios, hasta pronto!");  
}
```

Resolve conflicts before Merge



✓ All conflicted files have been resolved.

0 conflicted files

 ProyectoGitJava\src\Project\mainGitEjercicio.java
No conflicts remaining



[Open in command line](#), your tool of choice, or close to resolve manually.

Continue merge

Abort merge