



ED_T4_P1 : MANUAL DE USO DE UN GIT CLIENT

Eduardo Martín-Sonseca

Contenido

1. Instalar la herramienta elegida, en el caso de que sea necesario, y realizar la configuración inicial. ¿Por qué has elegido esta herramienta Git Client?	2
2. Crear un repositorio en GitHub que almacene el proyecto Java llamado AppGitClient, que contenga dos ramas: main y develop. El archivo MANIFEST.MD que contiene el proyecto debe ser ignorado en las copias instantáneas. Recuerda que debes crear el archivo README.MD con la descripción del proyecto. Clonar el repositorio en local.....	4
3. Crear un archivo llamado GitClient.java en la rama main y subirlo a remoto	8
4. Modificar el archivo anterior en remoto, insertando las siguientes líneas de código dentro de la clase Gitclient.java. Incluir el nuevo archivo en la rama develop.	10
5. Ver las diferencias del archivo anterior en local.	11
6. Actualizar el archivo anterior en local.....	11
7. Eliminar el archivo AppGitclient.java en local en la rama main.....	11
8. Ver el historial de confirmaciones.....	12
9. Actualizar todas las confirmaciones en remoto en la rama main.....	12
10. Crear una nueva rama en local llamada features/newrequest[1].	12
11. Crea un nuevo archivo llamado ClientGUI.java en la rama anterior.....	13
12. Fusionar los cambios de la rama anterior en la rama develop en local.....	14
13. Subir los cambios realizados a la rama develop en remoto.....	15

1. Instalar la herramienta elegida, en el caso de que sea necesario, y realizar la configuración inicial. ¿Por qué has elegido esta herramienta Git Client?

En esta práctica, he utilizado la herramienta GitHub Desktop, para instalarla, nos dirigiremos a nuestro navegador favorito y escribiremos en la barra de direcciones

<https://desktop.github.com/>, y clicaremos en **Download for Windows (64bit)**, mostrado en la **Ilustración 1**



Ilustración 1

Una vez descargado (Se instalará en nuestra carpeta de **Descargas**), se iniciará, y nos aparecerán 3 opciones que se mostrarán en la **Ilustración 2**

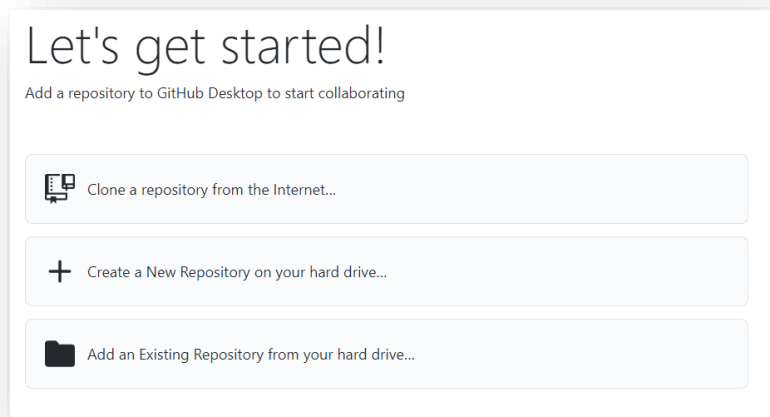


Ilustración 2

Una vez clicado en una de las tres opciones, tendremos que iniciar sesión en GitHub para clonar un repositorio, en este caso, una vez elegido el inicio de sesión, nos redijera automáticamente a la página web de GitHub.com para enlazar nuestra cuenta, mostrado en la **Ilustración3-4**

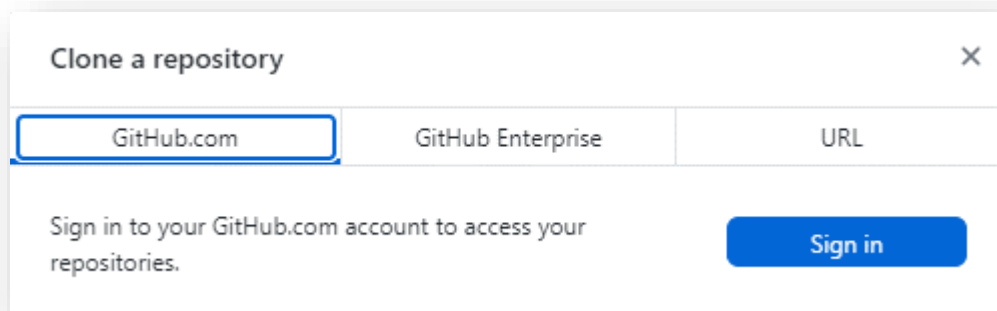
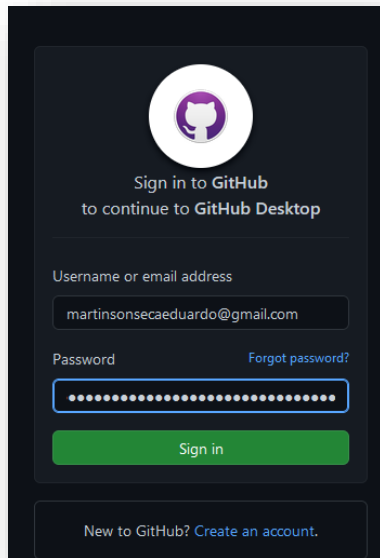
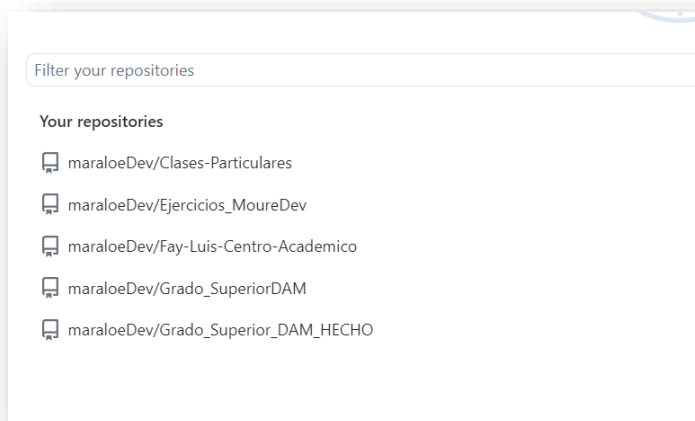


Ilustración 3

*Ilustración 4*

Una vez enlazada la cuenta a GitHub Desktop, ya podremos clonar el repositorio que queramos a cualquier parte del equipo

*Ilustración 5*

He elegido esta herramienta debido a que es muy rápida, y sencilla de utilizar, además, es totalmente gratuita, con una GUI sencilla para la administración de código sin necesidad de comandos, también es debido, a que si un repositorio, tiene varas ramas, se puede cambiar fácilmente de rama y de repositorio.

2. Crear un repositorio en GitHub que almacene el proyecto Java llamado AppGitClient, que contenga dos ramas: main y develop. El archivo MANIFEST.MD que contiene el proyecto debe ser ignorado en las copias instantáneas. Recuerda que debes crear el archivo README.MD con la descripción del proyecto. Clonar el repositorio en local.

Para crear un repositorio, nos dirigiremos a [GitHub](#), una vez ahí, clicaremos en **New**, mostrado en la Ilustración 6

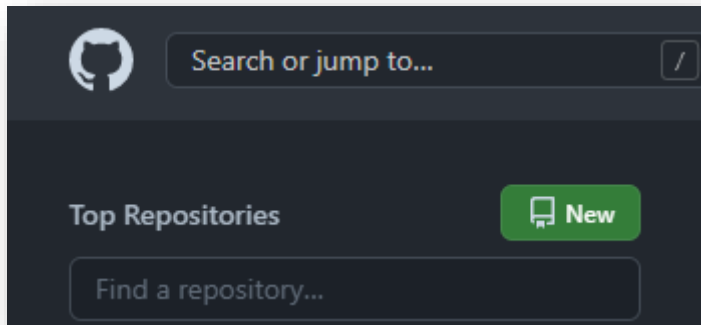


Ilustración 6

Una vez clicado, ingresaremos el nombre del nuevo repositorio a crear, en la ilustración 7, se mostrarán los campos rellenados, una vez creado, clicaremos en **Create repository**

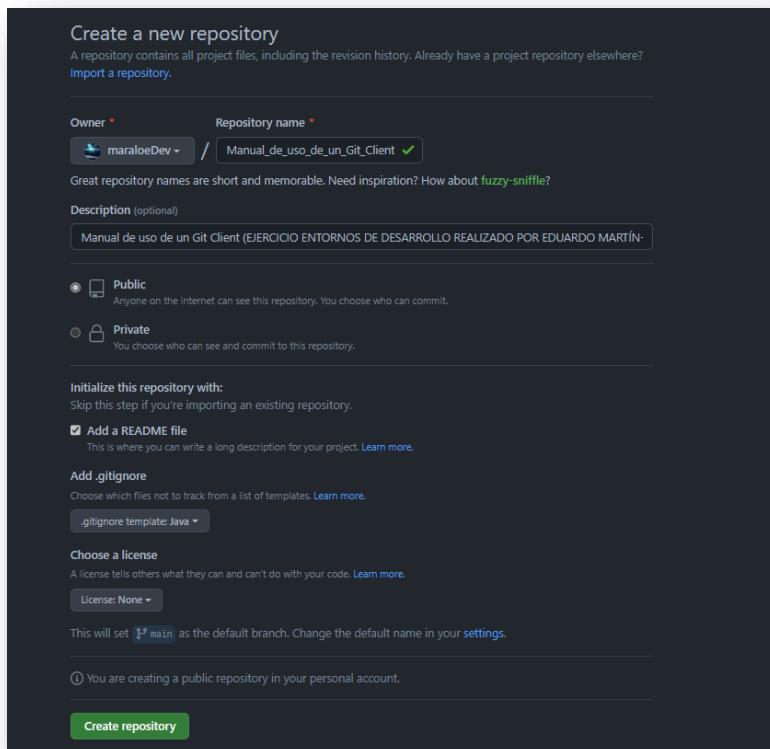


Ilustración 7

Una vez creado, almacenamos el proyecto **AppGitClient**, clicando en **Add file > Upload files**

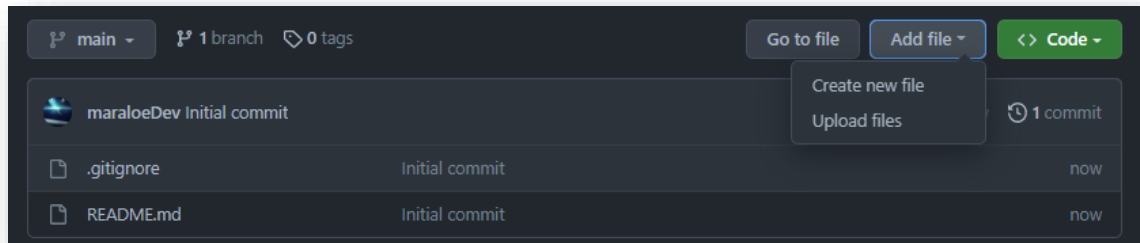


Ilustración 8

Una vez clicado, arrastraremos el proyecto, para que se agregue al repositorio, mostrado en la **Ilustración 9**

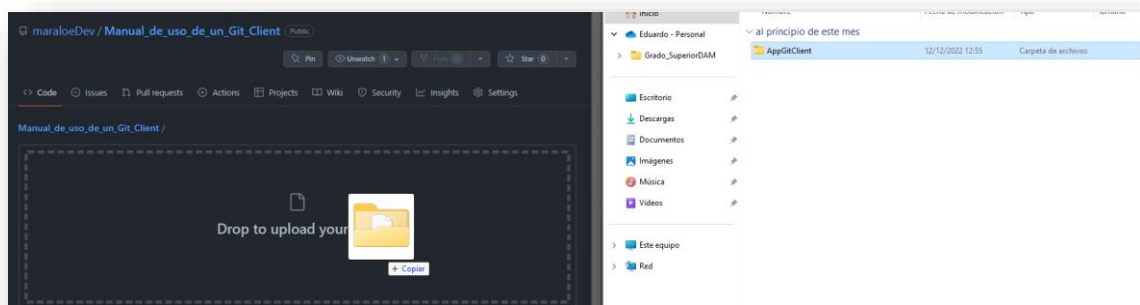


Ilustración 9

Una vez subidos los archivos, realizaremos una breve descripción de lo subido, mostrado en la **Ilustración 10**

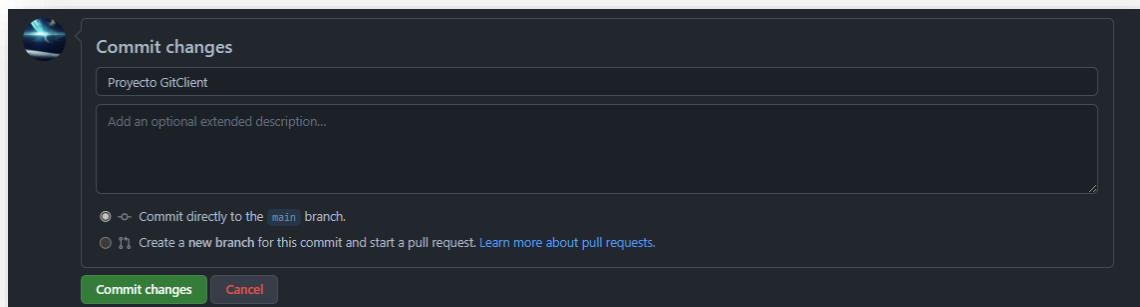


Ilustración 10

Una vez commiteado, crearemos la rama **develop**, para ello, nos dirigiremos a la pestaña **main**, y escribiremos el nombre de la rama, y clicaremos en **Create Branch : develop** from 'main', mostrado en la **Ilustración 11**

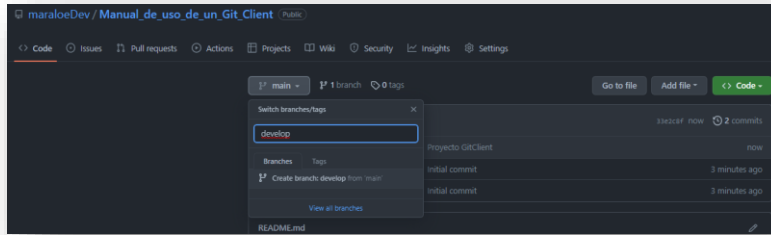


Ilustración 11

Para que un archivo que ignorado de las copias instantáneas, clicaremos en el archivo **.gitignore** para editarlo, y añadiremos el nombre del archivo a ignorar, mostrado en la **Ilustración 12**

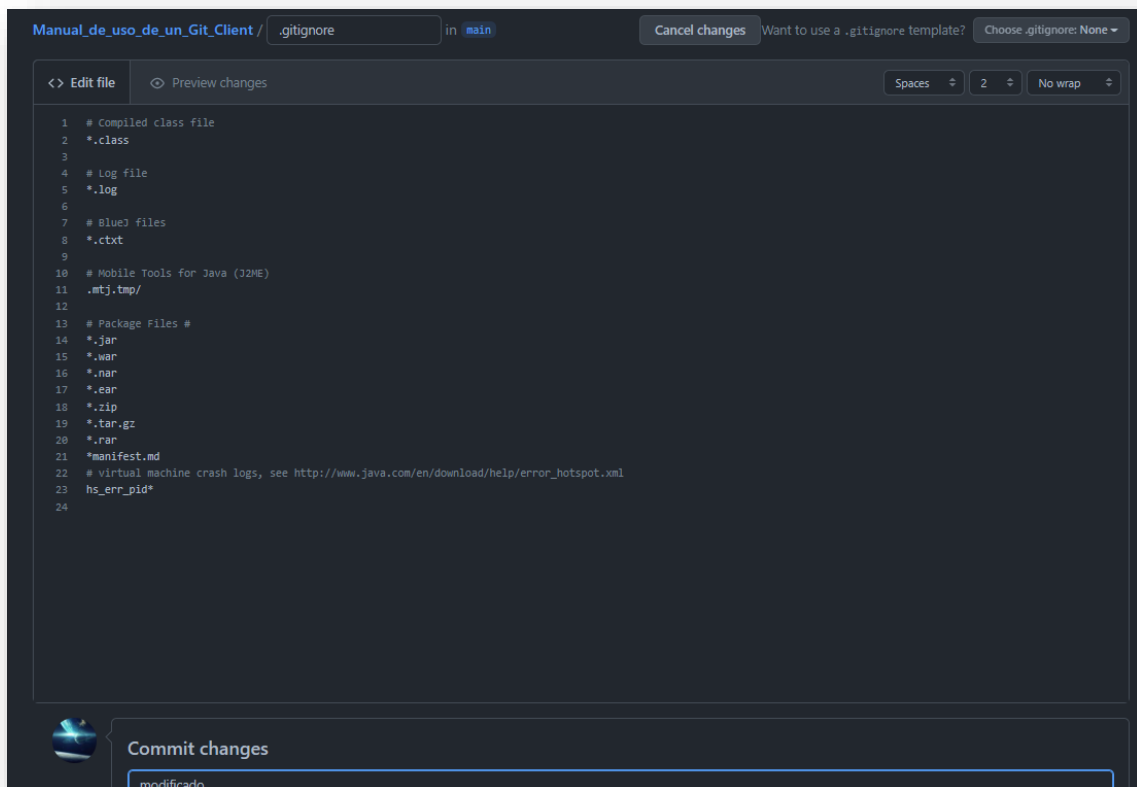


Ilustración 12

Para modificar el archivo README.MD, clicaremos en él, y añadiremos la descripción del proyecto, mostrado en la **Ilustración 13**

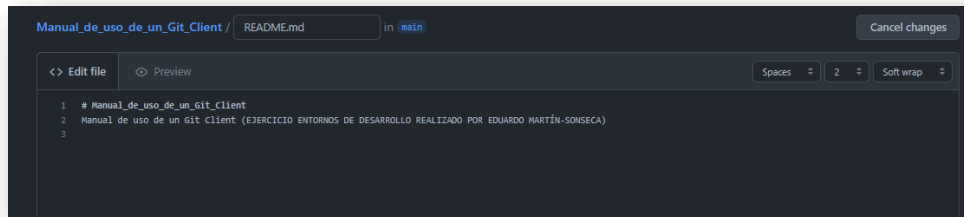


Ilustración 13

Para clonar el repositorio, nos dirigiremos a Git Hub desktop, y nos dirigiremos a la pestaña **File > Clone Repository Ilustración 14**

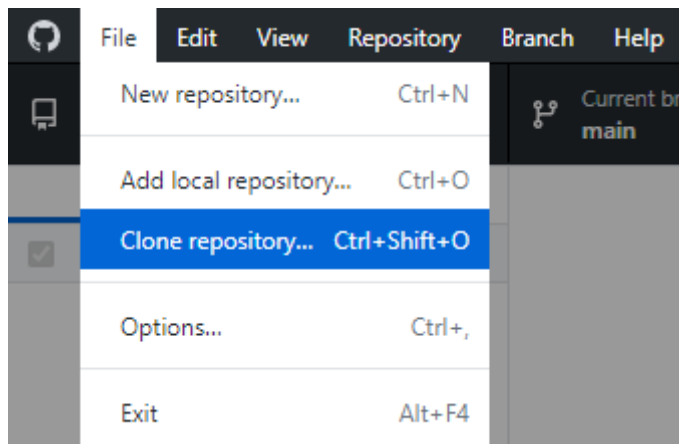
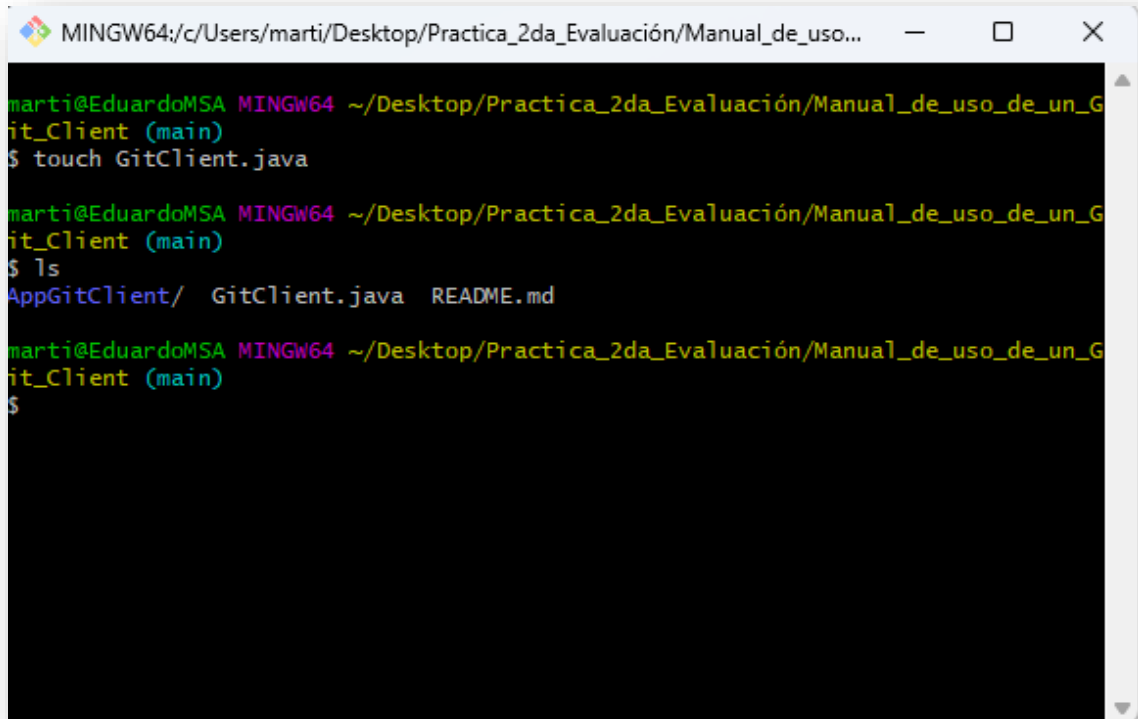


Ilustración 14

3. Crear un archivo llamado GitClient.java en la rama main y subirlo a remoto

Para crear un archivo en Git Bash, lo crearemos dentro del repositorio clonado anteriormente con **touch nombre del archivo**



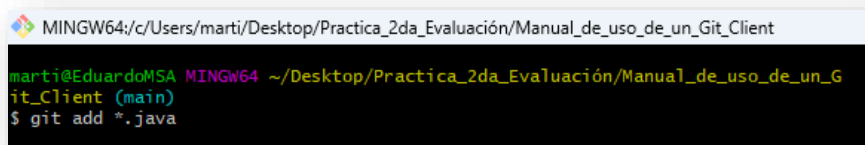
```
MINGW64:/c/Users/marti/Desktop/Practica_2da_Evaluación/Manual_de_uso_de_un_Git_Client
marti@EduardoMSA MINGW64 ~/Desktop/Practica_2da_Evaluación/Manual_de_uso_de_un_Git_Client (main)
$ touch GitClient.java

marti@EduardoMSA MINGW64 ~/Desktop/Practica_2da_Evaluación/Manual_de_uso_de_un_Git_Client (main)
$ ls
AppGitClient/  GitClient.java  README.md

marti@EduardoMSA MINGW64 ~/Desktop/Practica_2da_Evaluación/Manual_de_uso_de_un_Git_Client (main)
$
```

Ilustración 14

Una vez creado el archivo, lo subiremos a remoto mediante los siguientes comandos



```
MINGW64:/c/Users/marti/Desktop/Practica_2da_Evaluación/Manual_de_uso_de_un_Git_Client
marti@EduardoMSA MINGW64 ~/Desktop/Practica_2da_Evaluación/Manual_de_uso_de_un_Git_Client (main)
$ git add *.java
```

Ilustración 15

Una vez añadido, nos dirigiremos a gitHub Desktop, para commitarlo clicaremos en **Commit to main**

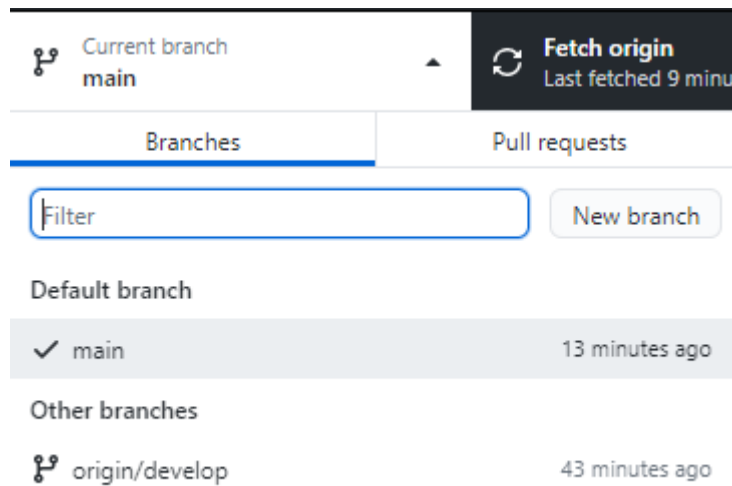


4. Modificar el archivo anterior en remoto, insertando las siguientes líneas de código dentro de la clase GitClient.java. Incluir el nuevo archivo en la rama develop.

```
@Override
public String toString() {
    return "GitClient{" +
        "cliente='" + cliente + '\'' +
        ", version='" + version + '\'' +
        ", sistemaOperativo='" + sistemaOperativo + '\'' +
        ", licencia='" + licencia + '\'' +
        '}';
}
```

Ilustración 16

Para incluirlo en la rama **develop**, nos dirigiremos a Current Branch, y clicaremos en la rama a cambiar



5. Ver las diferencias del archivo anterior en local.

Para poner la vista de las modificaciones en git hub desktop, nos dirigiremos al engranaje de la parte derecha, y clicaremos en **split**

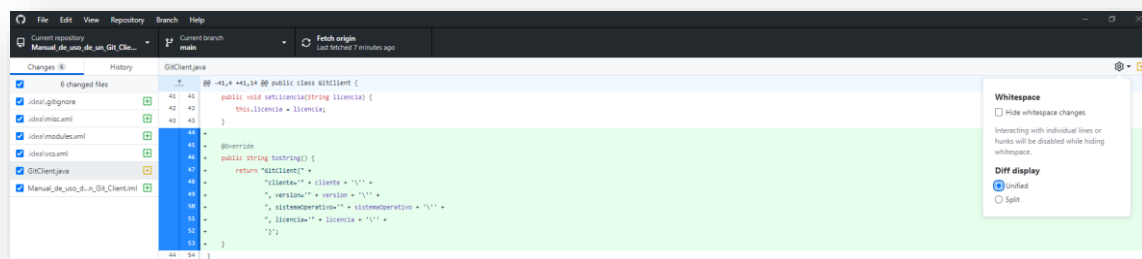


Ilustración 17

6. Actualizar el archivo anterior en local.

7. Eliminar el archivo AppGitclient.java en local en la rama main.

Para eliminarlo, nos dirigiremos a git hub, y clicaremos en el archivo, y lo eliminaremos

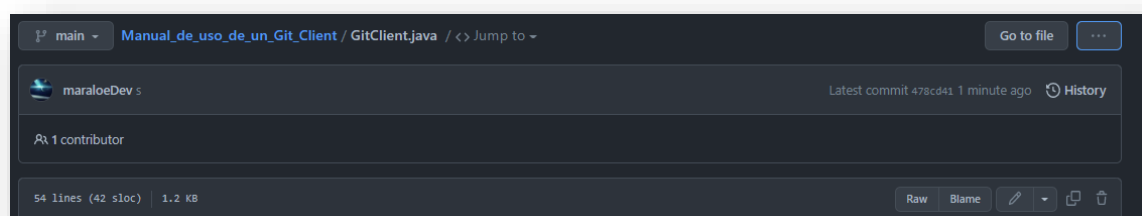


Ilustración 18

8. Ver el historial de confirmaciones.

Para ver el historial de confirmaciones, nos dirigiremos a githubDesktop, y clicaremos en **Hystory**

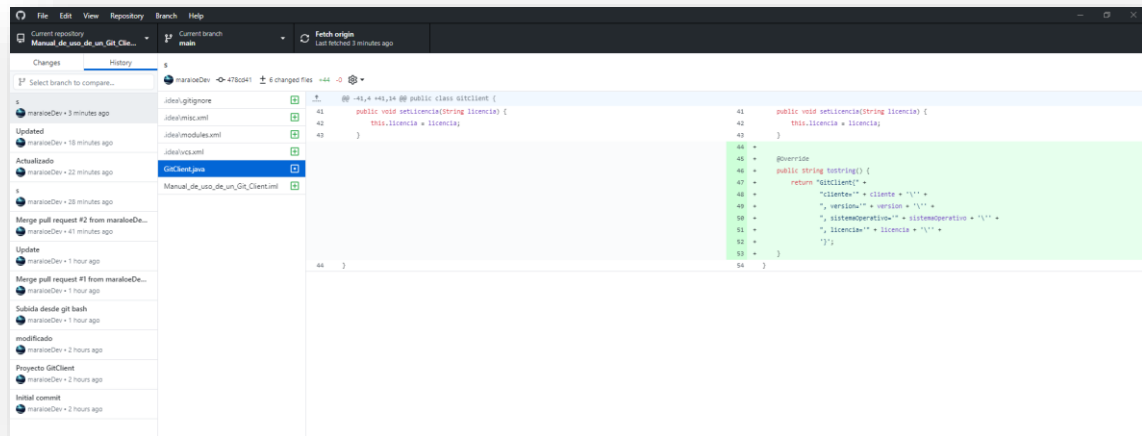


Ilustración 19

9. Actualizar todas las confirmaciones en remoto en la rama main.

10. Crear una nueva rama en local llamada features/newrequest[1].

Para crear una rama, nos dirigiremos a la pestaña **Branch > New Branch**

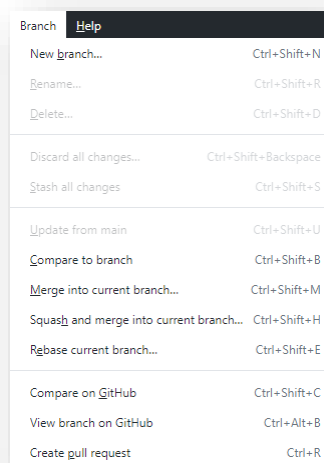


Ilustración 20

Una vez clicado, nos pedirá el nombre de la rama, y clicaremos en **Create Branch**

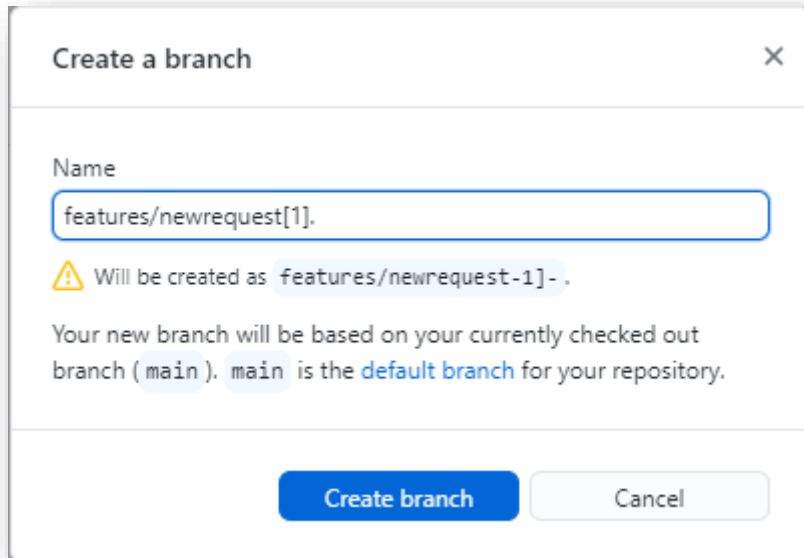


Ilustración 21

11. Crea un nuevo archivo llamado ClientGUI.java en la rama anterior

Para crear un nuevo archivo, nos dirigiremos a GitHub (cambiando a la rama creada anteriormente), y clicaremos en **Add file > Create new file**

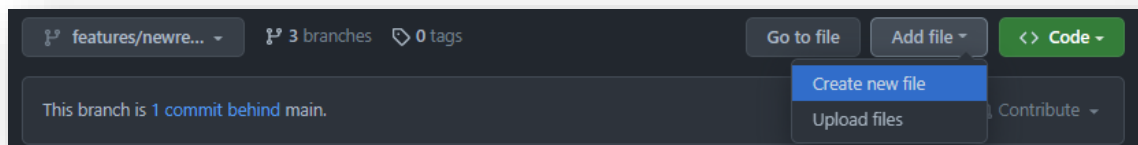


Ilustración 22

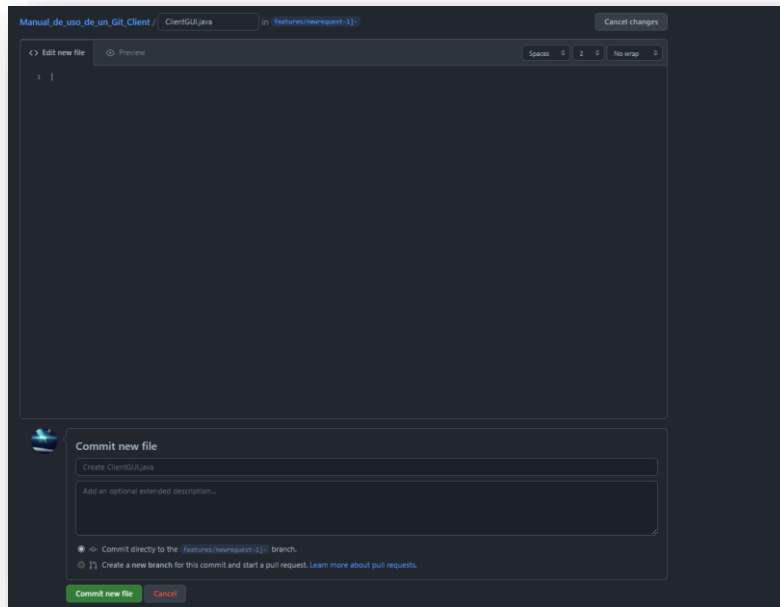
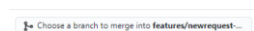
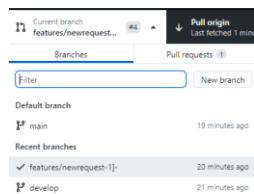


Ilustración 23

12. Fusionar los cambios de la rama anterior en la rama develop en local.

Para fusionar los cambios realizados, clicaremos en Current Branch, y seleccionaremos el botón que hay debajo,



Una vez clicado, nos aparecerá un cuadro de dialogo, en el que nos pedirá la rama a fusionar

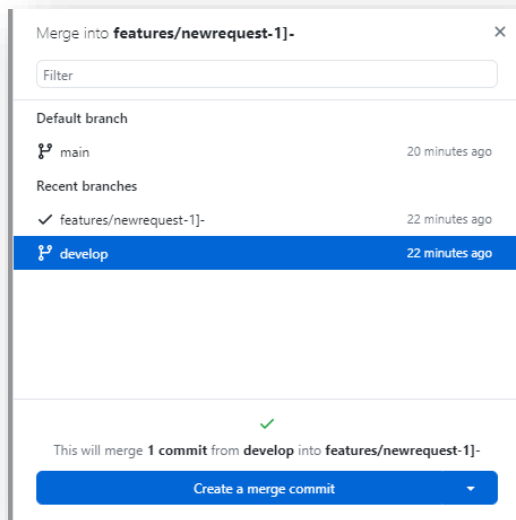


Ilustración 24

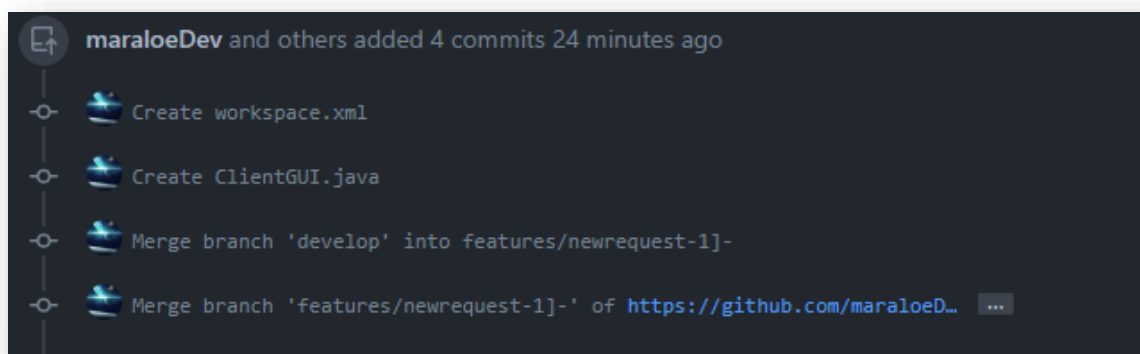


Ilustración 25

13.Subir los cambios realizados a la rama develop en remoto.