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](#_Toc121990582)

[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](#_Toc121990583)

[3. Crear un archivo llamado GitCliente.java en la rama main y subirlo a remoto 4](#_Toc121990584)

[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. 4](#_Toc121990585)

[5. Ver las diferencias del archivo anterior en local. 4](#_Toc121990586)

[6. Actualizar el archivo anterior en local. 4](#_Toc121990587)

[7. Eliminar el archivo AppGitclient.java en local en la rama main. 4](#_Toc121990588)

[8. Ver el historial de confirmaciones. 4](#_Toc121990589)

[9. Actualizar todas las confirmaciones en remoto en la rama main. 4](#_Toc121990590)

[10. Crear una nueva rama en local llamada features/newrequest[1]. 4](#_Toc121990591)

[11. Crea un nuevo archivo llamado ClientGUI.java en la rama anterior 4](#_Toc121990592)

[12. Fusionar los cambios de la rama anterior en la rama develop en local. 4](#_Toc121990593)

[13. Subir los cambios realizados a la rama develop en remoto. 4](#_Toc121990594)

# 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**

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

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**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

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**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 3

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 4

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

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

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.

# 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](https://github.com/) , una vez ahí, clicaremos en **New**, mostrado en la Ilustración 6

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

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**

Texto

Descripción generada automáticamente

Ilustración 7

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

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 8

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

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 9

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

Texto

Descripción generada automáticamente

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**

Captura de pantalla de un celular

Descripción generada automáticamente

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**

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración

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

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración

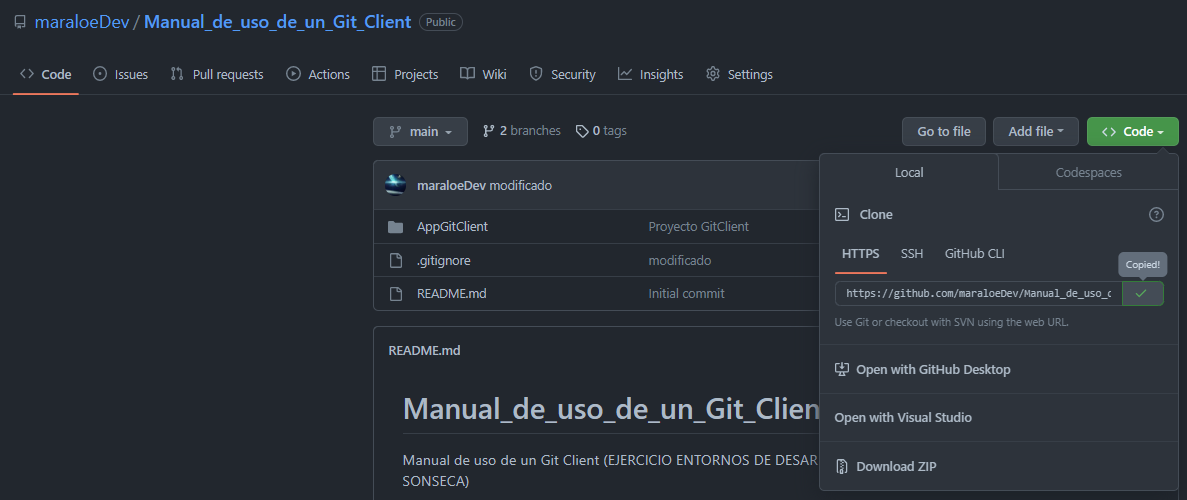
Para clonar el repositorio, nos dirigiremos a Git Bash, crearemos una carpeta en el escritorio, e inicializaremos el repositorio con el comando **Git init**, mostrado en la **Ilustración 14**

**Texto

Descripción generada automáticamente**

Ilustración

Una vez inicializado el repositorio, clonaremos el repositorio creado clicado en **Code > Local > HTTPS > copiar el link,** mostrado en la **Ilustración 15**

****

Ilustración

Una vez copiado, nos dirigiremos a Git Bash, e introduciremos el comando **git clone**, mostrado en la **Ilustración 16**

Texto

Descripción generada automáticamente

Ilustración

# 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**

Texto

Descripción generada automáticamente

Ilustración

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

Texto

Descripción generada automáticamente

Ilustración

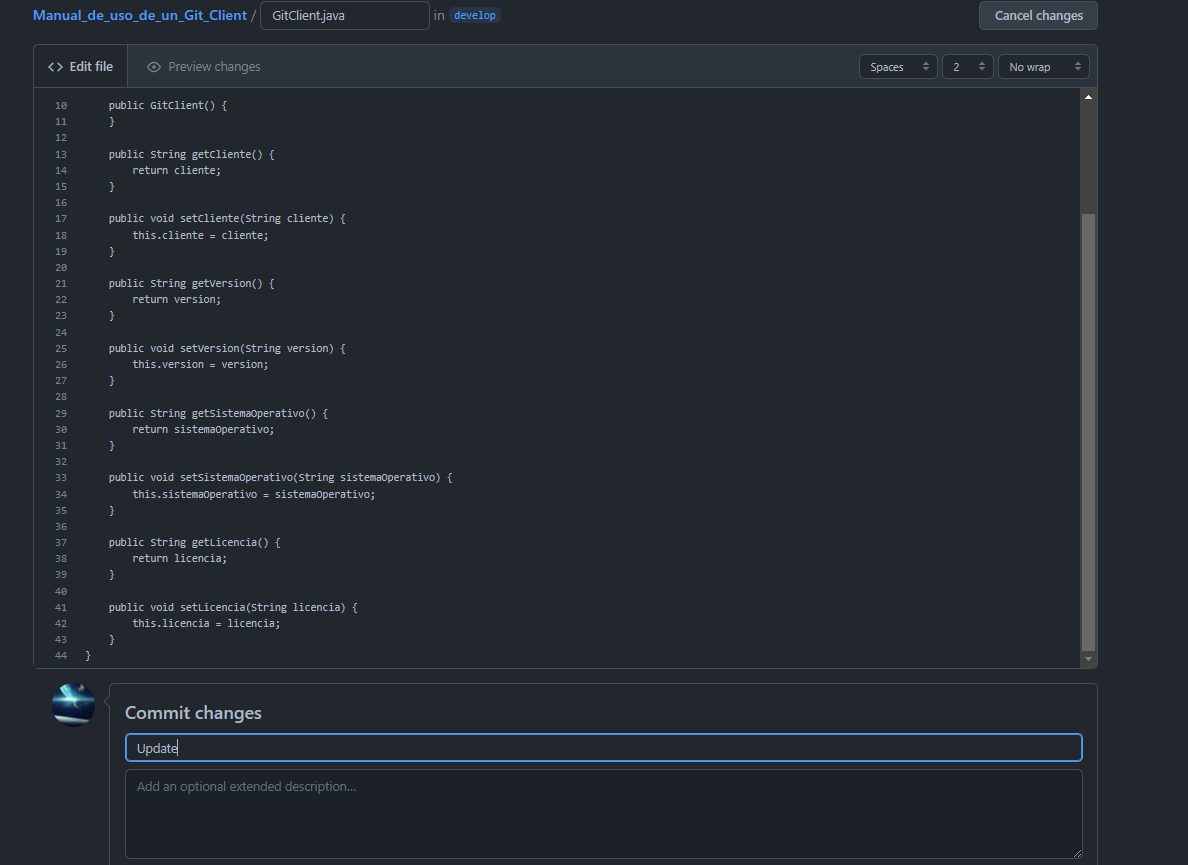
Una vez subido a remoto, comprobaremos que se ha commiteado correctamente

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración

# 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.



Ilustración

# Ver las diferencias del archivo anterior en local.

# Actualizar el archivo anterior en local.

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

# Ver el historial de confirmaciones.

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

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

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

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

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