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](#_Toc121951375)

[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](#_Toc121951376)

# 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.

# Crear un archivo llamado GitCliente.java en la rama main y subirlo a remoto

# 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.

# 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.