



## Introducción

En esta práctica aprenderemos a instalar, configurar y utilizar sistemas de control de versiones GIT y GITHUB.

Para realizar las versiones utilizaremos la refactorización del código.

Git es un software de control de versiones desarrollado por Linus Torvalds (el padre de linux). Uno de los propósitos de GIT es la eficiencia y confiabilidad de las distintas versiones de las que consta un proyecto software. Cuando dicho proyecto tiene un gran número de archivos, es necesario tener un software fiable y eficiente

## PRACTICA

### APARTADO 1

**Basándonos en el código Main.kt, realiza los siguientes ejercicios.**

1. Configura CVS para el proyecto **DAW\_ED04\_Actividad \_Nombre**. Crea un nuevo repositorio (deposito local) en caso de no disponer de él. Indicar que tienes que tener instalado para crear el repositorio.
2. Realiza las siguientes modificaciones (versiones) en el proyecto (Main.kt), comentando el resultado de la ejecución.
  - a. Utilizando refactorización modifica la variable suma2 por sumaColumnas. Guardar esta nueva versión mediante comandos.
  - b. Utilizando refactorización crea una función sumarFilas. Guardar esta nueva versión mediante el ED.
3. Muestra el historial de versiones para el fichero Main.kt (tanto en consola como en ED) y comenta el resultado.
4. Modifica el repositorio local, para que quede como se al principio de subirlo al repositorio.



## APARTADO 2

---

### 1. Crea un nuevo repositorio local

Crea un nuevo repositorio en la herramienta Git ya instalada anteriormente, con el nombre de “ED\_Nombre\_Apellido”.

### 2. Utiliza un proyecto ya existente.

Lo más normal es descargar (clonar) un proyecto ya existente de GitHub o crear una copia de algún proyecto que ya se disponga.

Realiza una clonación a tu equipo del siguiente proyecto de GitHub:

<https://github.com/twitter/twitter4j.git>

### 3. Añade ficheros al repositorio local

Crea o copia un nuevo fichero a la carpeta del repositorio (local)

Una vez copiado dicho fichero, actualiza los cambios en el repositorio. AL actualizar, etiquétalo como “primer commit”.

### 4. Visualiza el estado del repositorio

Visualiza el estado del repositorio para verificar si hay algún cambio que todavía no ha sido actualizado.

## APARTADO 3

---

### 1. Crea un nuevo repositorio en GitHub

En una cuenta de GitHub crea un repositorio llamado **Entornos\_nombreapellidos** para almacenar el código de un proyecto en kotlin (que tenga varias clases).

Indica todos los pasos que tienes que realizar para subir tu proyecto del repositorio local a github.

### 2. Añadir nuevos ficheros al repositorio.

- Añade un fichero README.md al repositorio con el título del proyecto, una descripción del mismo y la información del autor
- Añade un fichero .gitignore para ignorar los ficheros compilados del proyecto y otra carpeta que has creado con el tu nombre.
- Cuando tengas la primera funcionalidad terminada sube el código al repositorio.



### 3. Trabajar sobre el repositorio

Sobre el repositorio anterior, añade una nueva funcionalidad (un nuevo método, por ejemplo):

- Añade la funcionalidad requerida
- Modifica el README.md para añadir tu nombre como colaborador del mismo
- Comprueba el estado del repositorio en tu equipo (status)
- Añade los cambios realizados al proyecto

### 4. Trabajar sobre el repositorio con ficheros

- Sobre el mismo repositorio que los puntos anteriores, debes de eliminar un fichero que ya no es necesario que forme parte del mismo (si hace falta añade alguno para luego eliminarlo) y tampoco quieres que siga en tu equipo.

### 5. Ampliación - Wiki

Añade ahora en la Wiki del proyecto información sobre como instalar, ejecutar y usar la aplicación (3 páginas diferentes enlazadas y organizadas desde la portada de la Wiki).

## Entrega

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. Con las capturas e indicaciones pertinentes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas: UD4\_ED\_Nombre\_Apellidos.pdf

## Calificaciones

Criterios de calificación	Hecho	Calificación
Apartado 1 (4puntos)-		
Apartado 2 (2 puntos)-		
Apartado 3(4 puntos) + 0,5 (ampliación)		