**UT4 - Optimización y Documentación**

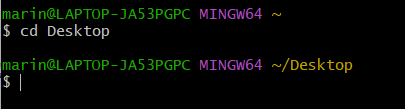
**——————————————————————————————**

**Apartado 1**

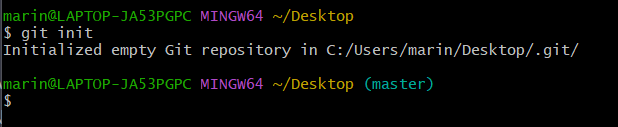
1. **Configura CVS para el proyecto DAW\_ED04\_Actividad\_Nombre. Crea un nuevo repositorio (deposito local) en caso de no disponer de él. Indicar que tienes que tener instalado para crear el repositorio.**

Abrimos Git Bash.

En la terminal accederemos al directorio donde queramos crear el repositorio de la siguiente manera.

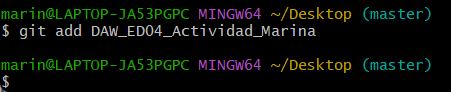


Una vez echo esto ejecutaremos el comando “git init”, esto lo que hará es crear un nuevo repositorio vacío en la ubicación deseada.



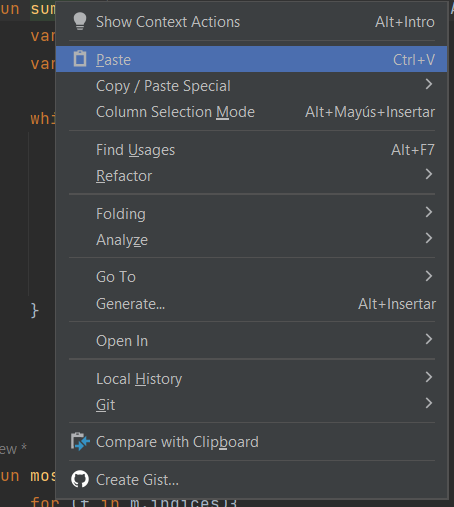
De esta forma ya tendremos creado nuestro repositorio.

Ahora añadiremos el archivo DAW\_ED04\_Actividad\_Nombre con el comando “git add” seguido del nombre del archivo o carpeta que deseemos añadir.

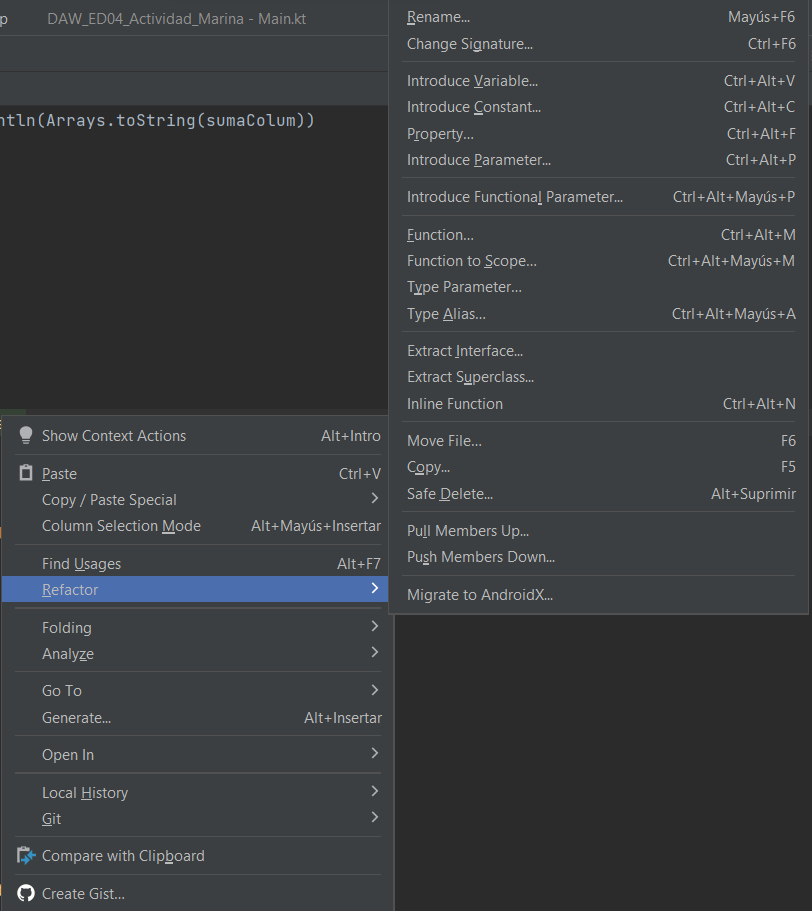


1. **Realiza las siguientes modificaciones (versiones) en el proyecto (Main.kt), comentando el resultado de la ejecución.**
2. **Utilizando refactorización modifica la variable suma2 por sumaColumnas. Guardar esta nueva versión mediante comnados.**

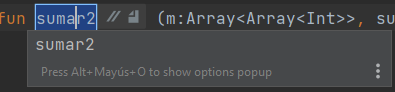
Colocaremos el cursor en la función de suma2 y daremos click derecho, esto nos mostrara lo siguiente.



Aquí seleccionaremos refactor y veremos otro menú al lado donde encontraremos Rename (nos sale en la primera posición).

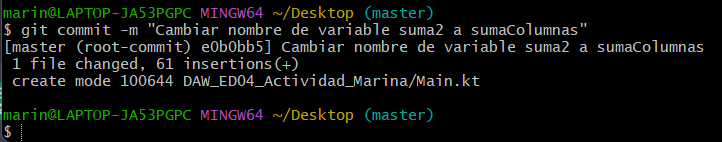


Una vez que le demos a Rename veremos que podremos cambiar el nombre de la función.



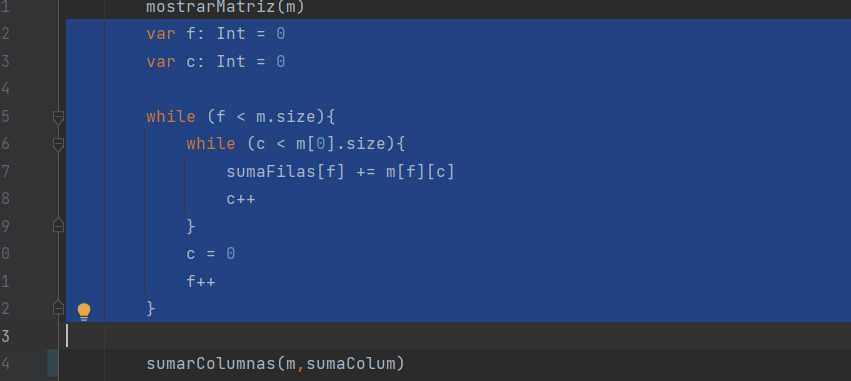
Una vez tengamos el nombre que deseemos solamente tendremos que dar a Enter.

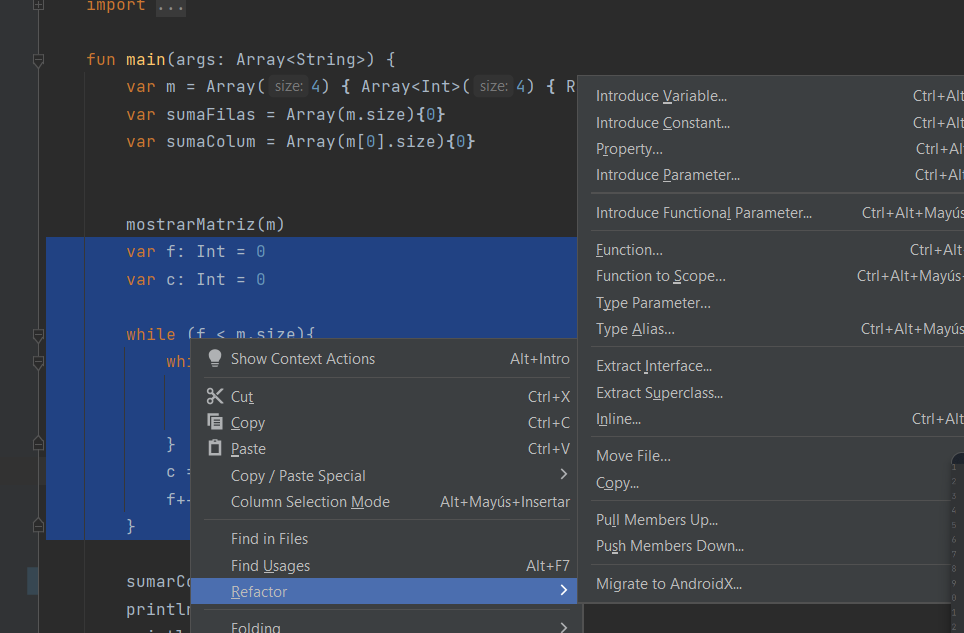
A continuación, guardaremos los cambios por comandos. Para ellos utilizaremos el comando -- git commit -m “Cambiar nombre de variable suma2 a sumaColumnas”.



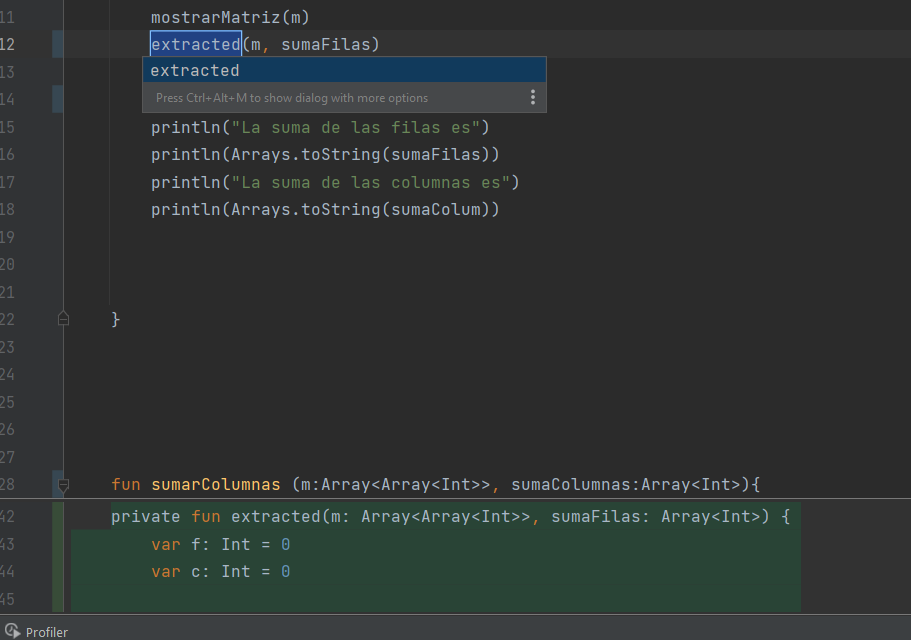
1. **Utilizando refactorización crea una función sumarFilas. Guardar esta nueva versión mediante el ED.**

En el main, dentro del while es donde se suman las filas seleccionaremos desde la línea 12 a la 22, aquí daremos click derecho y en el menú refactor y seleccionaremos Function.

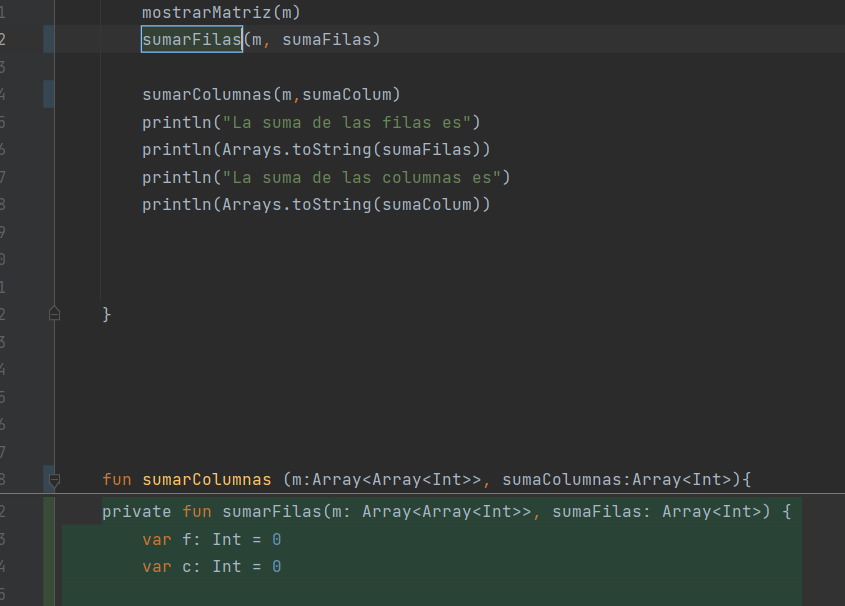




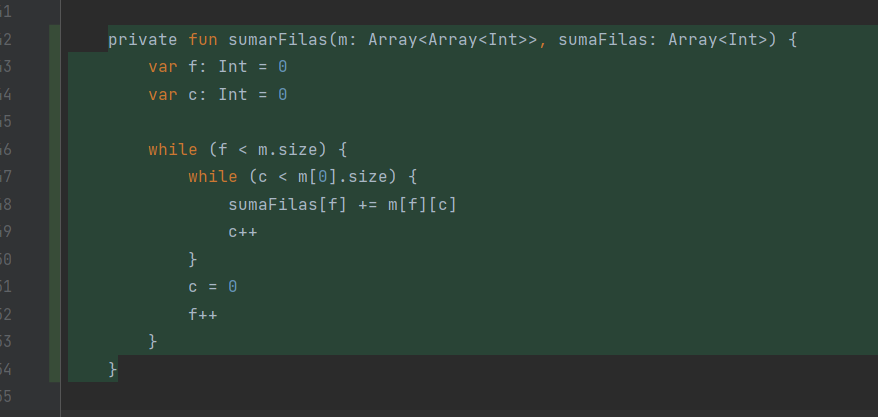
Esto nos creara lo siguiente.



Aquí pondremos el nombre que deseemos que tenga nuestra función

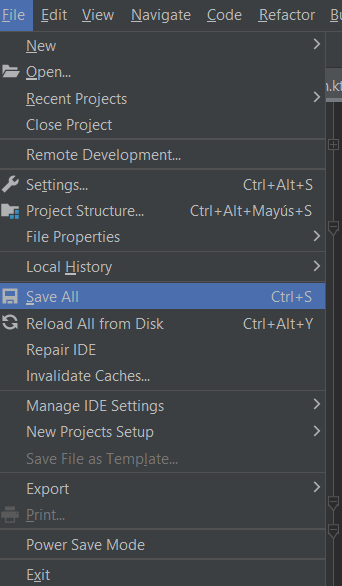


Y ya tendremos la función creada.



A continuación, vamos a guardar estos cambios desde el ED (entornos de desarrollo integrado), Guardar nuestro proyecto desde aquí es muy fácil, tenemos dos opciones para guardarlo.

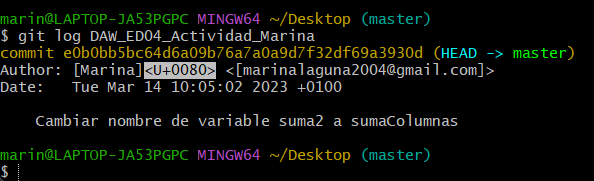
La primera opción que vamos a tener es en el menú que tendremos arriba encontraremos la opción de FILE, esta la seleccionaremos y nos abrirá el siguiente menú.



Como lo que nos interesa es guardar nuestro proyecto seleccionaremos Save All para guardarlo y como vemos de la otra forma que podemos guardarlo es dando Ctrl + S.

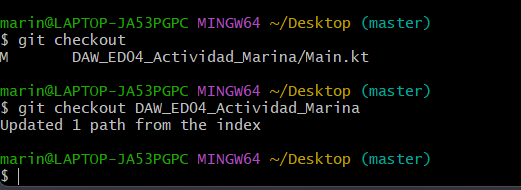
1. **Muestra el historial de versiones para el fichero Main.kt (tanto en consola como en ED) y comenta el resultado.**

Para mostrar el historial de versiones en consola, podemos hacerlo mediante el siguiente comando “git log”, esto mostrará lo siguiente



1. **Modifica el repositorio local, para que quede como al principio de subirlo al repositorio.**

Para volver al estado original del repositorio local, utilizaremos el comando “git checkout” seguido del nombre del archivo que hay que restaurar.



**Apartado 2**

1. **Crea un nuevo repositorio local con el nombre de “ED\_Nombre\_Apellido”.**

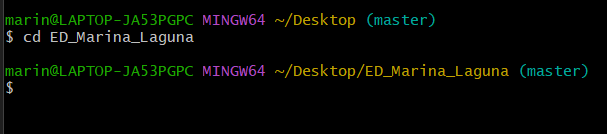


1. **Utiliza un proyecto ya existente. Lo más normal es descargar (clonar) un proyecto ya existente de GitHub o crear una copia de algún proyecto que ya se disponga.**

**Realiza una clonación a tu equipo del siguiente proyecto de GitHub:** <https://github.com/twitter/twitter4j.git>

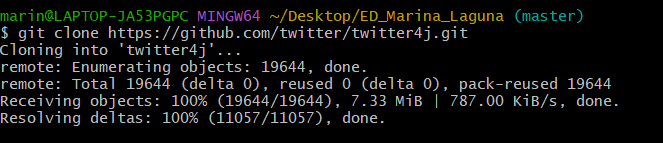
Para hacer la clonación del proyecto, desde la terminal de Git Bash haremos lo siguiente:

Lo primero que necesitamos es estar en la ubicación de nuestro repositorio donde queremos realizar la clonación.



Una vez que tengamos la ubicación deseada, procederemos ha hacer la clonación del proyecto de GitHub de la siguiente manera:



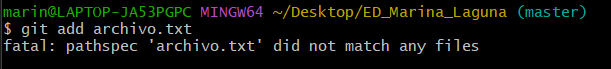


De esta forma se realizará la clonación que deseamos.

1. **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”.**

Deberemos asegurarnos de que nos encontramos en el repositorio local en la terminal.

Una vez que sepamos cual es el archivo que queremos copiar, lo haremos de la siguiente manera.



Para etiquetarlo ejecutaremos el siguiente comando

git commit -m "primer commit"

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

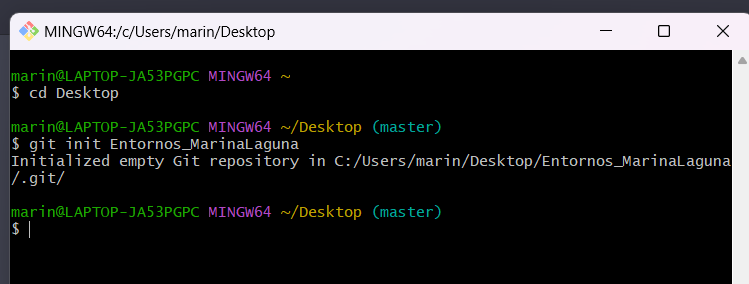
Para visualizar el estado en el que se encuentra nuestro repositorio podemos ejecutar el siguiente comando.

git status

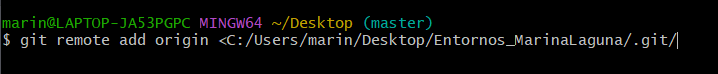
esto nos mostrara información sobre los cambios que se hayan realizado en el repositorio y su estado actual

**Apartado 3**

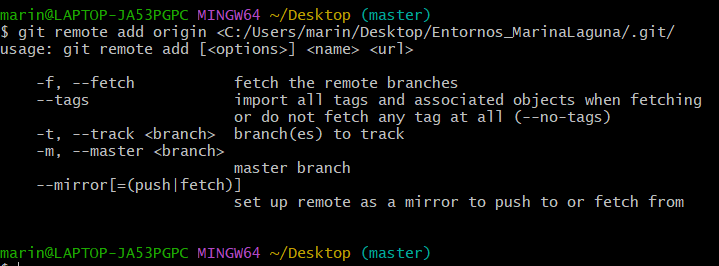
1. **Crea un nuevo repositorio en GitHub. En una cuenta de GitHub crea un repositorio llamado Entornos\_nombreapellidos para almacenar el código de un proyecto en kotlin (que tenga varias clases). Indica todos los pasos que tienes que realizar para subir tu proyecto del repositorio local a github.**



Para conectar mi repositorio local ejecutaremos el siguiente comando:



Nos devolverá lo siguiente



Ahora comprobaremos si está correcta la conexión con el siguiente comando:



Para subir los cambios al repositorio remoto usaremos el siguiente comando:



1. **Añadir nuevos ficheros al repositorio.**
2. **Añade un fichero README.md al repositorio con el título del proyecto, una descripción del mismo y la información del autor.**

Lo primero debemos estar ubicados en el repositorio local que vamos a crear el fichero.



Para añadir un fichero README.md lo haremos de la siguiente manera:



Para guardar los cambios realizaremos un commit

git commit -m “Agregado archivo README.md”

Para subir los cambios ejecutaremos este comando

Git push origin master

1. **Añade un fichero. gitignore para ignorar los ficheros compilados del proyecto y otra carpeta que has creado con el tu nombre.**
2. **Cuando tengas la primera funcionalidad terminada sube el código al repositorio.**
3. **Trabajar sobre el repositorio.**

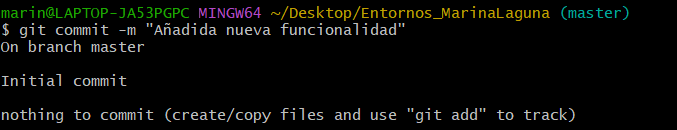
**Sobre el repositorio anterior, añade una nueva funcionalidad (un nuevo método, por ejemplo):**

1. **Añade la funcionalidad requerida.**

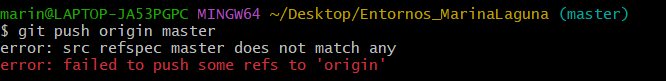
Para añadir la funcionalidad ejecutaremos el siguiente comando



Ahora realizaremos un commit con los cambios



Y para subirlos haremos lo siguiente



1. **Modifica el README.md para añadir tu nombre como colaborador del mismo.**
2. **Comprueba el estado del repositorio en tu equipo (status).**

Para comprobar el estado del repositorio utilizamos el comando git status.

1. **Añade los campos realizados al proyecto.**
2. Trabajar sobre el repositorio con ficheros.
3. 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.
4. Ampliacion – wiki

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