**Comandos básicos para usar la consola de Git.**

Si es la primera vez que usaremos el Git, tendremos que introducir en la consola nuestro Usuario y Email y luego hacer commit.

**git config –global user.name “Juan” (entre comillas):** Con este comando lo que hacemos es introducir un usuario de trabajo, si es la primera vez que vamos a usar la consola de git.

**git config –global user.email**[**informatica.ruizdiaz@gmail.com**](mailto:informatica.ruizdiaz@gmail.com) **(sin comillas ni paréntesis):** Con este comando lo que hacemos es introducir un email de trabajo, si es la primera vez que vamos a usar la consola de git.

Ejemplo:

git config --global user.email [you@example.com](mailto:you@example.com)

git config --global user.name "Your Name"

**git init “documento/carpeta”:** Inicia el respaldo/seguimiento de archivos o carpetas. Crea 2(dos) áreas de trabajo,

“1 - Área de ensayo (staging area)”,

“2 – Repositorio local”.

**git add “documento/archivo”:** Este comando realiza seguimiento al archivo/carpeta el cual le indicamos. Además, con este comando lo que conseguimos es que git lleve nuestros archivos/carpetas al área de ensayo.

**git commit -m “Nombre de la modificación” (entre comillas):** Con este comando lo que git realiza es trasladar desde el área de trabajo al repositorio Local. Y aquí se crea el respaldo del archivo/carpeta.

**git status -s:** Comando para saber en qué estado se encuentra nuestro archivo/carpeta y a partir de ahí tomar una decisión de que hacer.

**git log –oneline:** Para saber todo el listado de las copias archivos/carpetas con los códigos correspondientes.

**git reset - -hard:** Con esto lo que conseguimos es una restauración del archivo sin tener en cuenta ningún punto.

**git reset - -hard “Código”:** Con este código lo que se logra es volver a un punto anterior del archivo.

“**Git add .” :** Aquí lo que hacemos es que le decimos a git que agregue y haga seguimiento a todos los archivos/carpetas que pudiera a ver en el área de “1 - Área de ensayo (stagingarea)”.

**git commit -am “descripción de lo que se realizó”:** Este comando se utiliza para que git guarde varios archivos/documentos a la vez, que fueron modificados.

Editor **“vim”:** Sirve para modificar la descripción del documento que hemos guardado, para acceder a eso se usa el **comando “git commit–amend”,** se abre el editor.

Para empezar a editar se introduce el comando “i” + enter, luego se puede empezar a editar el archivo.

Se Borra el nombre del archivo, luego Esc, : (dos puntos) y se empieza a escribir la descripción correcta-

Luego de modificar el archivo pulsamos las teclas**wq** para guardar y salir del editor.

**git remote add origin** [**https://github.com/RDINFORMATICARUIZDIAZ/Proyecto\_web.git**](https://github.com/RDINFORMATICARUIZDIAZ/Proyecto_web.git)**: Comando para subir archivo existente en Github**

Si quieres comprobar tu configuración, puedes usar el comando **gitconfig --list**para mostrar todas las propiedades que Git ha configurado:

Para alzar archivos/documentos de local al repositorio se llama PUSH

Y para el inverso se llama PULL.

**Git tag 15-03-23v1 -m “versión 1 del proyecto”:** este comando se utiliza para guardar la versión del proyecto terminado.

**Git push- -tags:** Este comando sirve para alzar la versión en remoto

**Git clone** [**https://github.com/RDINFORMATICARUIZDIAZ/Relog\_Digital.git**](https://github.com/RDINFORMATICARUIZDIAZ/Relog_Digital.git) **:** sirve para clonar documentos/archivos de remoto a local

Ramas o Branch

Ramas o Branch (Master): línea de Tiempo, puede existir muchas ramas o Branch

**Git branch “nombre que quieras poner (sin comillas)”:** este comando sirve para crear otra rama o branch en consola.

**Git branch:** sirve para ver todas las Ramas actúales.

**git checkout “nombre de la rama en la que me quiero mover (sin comillas)”:** Este comando sirve para moverse de una rama a otra.

**Para realizar un march**

1 se debe posicionar en la rama master

**Git merge** nombre con lo que se quiere fusionar

**Git Branch -d** “nombre de la rama que se quiera borrar” (sin comillas):