CONFIGURACION DE USUARIO

$ git config --global user.name "[name]" # Genera un nombre de usuario global.

$ git config --global user.email "[email address]" # Genera un nombre de usuario global.

$ git config user.name "[name]" # Genera un nombre de usuario para un repositorio particular.

$ git config user.email "[email address]" # Genera un nombre de usuario para un repositorio particular.

REPOSITORIOS

$ git init [project-name] # Inicia un nuevo repositorio.

$ git clone [url del proyecto] # Para descargar y sincronizar por primera vez un proyecto.

$ git remote add origin [url del repositorio remoto] # Sirve para sincronizar nuestro repositorio local con un repositorio remoto de Github.

$ git remote -v # Para comprobar si nuestro repositorio local está sincronizado con algún repositorio remoto creado en Github.

HACER CAMBIOS Local

$ git status # Permite ver el estado de nuestros archivos, de las ediciones y cambios.

$ git diff # Muestra las diferencias.

$ git add . # Envía todos los archivos al área de preparación.

$ git add (nombre del archivo.extensión) Envía solo el archivo referenciado al área de preparación.

$ git commit -m "[descriptive message]" # Guarda un parque con los cambios que hemos hecho, con un mensaje.

USO DE LAS BRANCH

$ git branch [branch-name] # Crea una rama.

$ git branch # Enumera todas las ramas de tu repositorio, es similar a git branch –list.

$ git checkout [branch-name] # Te ubicas en esa rama. Generalmente, Git solo permitirá que nos movamos a otra rama si no tenemos cambios. Si tenemos cambios, para cambiarnos de rama, debemos: 1. eliminarlos (deshaciendo los cambios); 2. conﬁrmarlos (haciendo un git commit).

$ git merge [branch] # Mergea una rama.

$ git branch -d [branch-name] # Borra la rama, Git evita que eliminemos la rama si tiene cambios que aún no se han fusionado con la rama Main.

$ git branch -D [branch-name] # Fuerza la eliminación de la rama especiﬁcada, incluso si tiene cambios sin fusionar.

HISTORIAL

$ git log # Muestra el historial.

$ git log --oneline # Mostrar información en una sola línea.

$ git log -[n] # Ver un número limitado de commits.

$ git log -p # Ver información extendida del commit.

$ git log --all --decorate --oneline –graph # Ver información de los logs a color y en una sola línea.

$ git diff [first-branch]...[second-branch] # Muestra las diferencias entre las ramas.

SINCRONIZAR CAMBIOS

$ git push origin [branch] # Envia los commits a la rama que nombremos.

$ git pull # Descarga y actualiza tu repositorio local con todo lo que está en el repositorio remoto de Github.

$ git pull origin “nombre rama” # Descarga y actualiza la rama especificada.

$ git rev-parse --show-toplevel.