Git Local:

Git init: Iniciar un repositorio vacío en una carpeta especifica.

Git clone [url] [dir]: Obtenemos un repositorio.

Git status: Revisamos el estado de los archivos.

Git add ‘nombre\_ de\_archivo’: Añadir un archivo específico al área de preparación.

Git add . : Añadir todos los archivos del directorio.

Git commit –m “mensaje”: Confirmar los cambios realizados.

Git log: Muestra el historial de confirmaciones.

Git diff: Muestra los cambios con respecto a la última versión guardada en el repositorio.

Renombrar archivos y directorios

Para poder renombrar/mover un archivo y/o directorio y que dicho cambio ya esté en el área de preparación podemos usar el comando:   
git mv <old\_filename><new\_filename>  
git mv <old\_dirname><new\_filename>

Si borramos manualmente veremos que aparece una modificación: archivo borrado. Y también aparece un archivo nuevo sin seguimiento

Eliminar archivos y directories

Para poder eliminar un archivo y/o directorio y que dicho cambio ya esté en el área de preparación podemos usar el comando:

git rm   
git rm /\*  
git rm –r /  
Si queremos que sólo se elimine del repositorio de git pero permanezca en nuestro directorio de trabajo debemos utilizar la opción  
--cached   
git rm –cached

Con el comando rm nos queda listo para confirmar:   
Si borramos manualmente veremos que aparece una modificación: archivo borrado.

.gitignore

También acepta expresiones regulares como   
‘nombre\_de\_carpeta/\*.txt’

Configurar rama local con rama remota υ Alternativas υ Subiendo las confirmaciones   
git push –u origin main   
git push --set-upstream origin main   
Sin subir las confirmaciones   
git branch –u origin/main   
git branch --set-upstream-to=origin/main

Alternativas υ Subiendo las confirmaciones   
git push –u origin main   
git push --set-upstream origin main   
Sin subir las confirmaciones υgit branch –u origin/main   
git branch --set-upstream-to=origin/main  
  
git branch –vv nos listará los nombres de las ramas locales y la rama remota con la que está enlazada cada una de ellas.

¿Qué sucede si alguien ha realizado cambios antes que nosotros?   
¿Qué deberíamos hacer?

Decidimos conservar ambos cambios

υConfirmamos la resolución de los conflictos con un nuevo commitυConfirmamos la resolución de los conflictos con un nuevo commit

υAhora estamos adelantados por dos commits

υ Finalmente, estando actualizados con el repositorio remoto y habiendo resuelto los respectivos conflictos podemos subir los nuevos cambios al repositorio

υ Ver repositorios remotos configurados υgit remote υ

Añadir repositorio remoto υgit remote add [nombre] [url] υ

Remover repositorio remoto υgit remote rm [nombre] υ

Ver ramas existentes en el repositorio υgit branch υ

Subir confirmaciones a repositorio remoto υgit push [nombre-remoto] [rama] υ

Actualizar de repositorio remoto υgit pull [nombre-remoto] [rama]

Git remoto

Git remote: ver repositorios remotos configurados.

Git remote add [nombre] [url]: Añadir repositorio remoto.

Git remote rm {nombre]: Remover el archivo.

Git branch: Ver ramas existentes.

Git push [nombre-remoto] [rama]: Subir confirmaciones.

Git pull [nombre-remoto] [rama]: Actualizar los cambios.

Git branch

Git branch: lista las ramas locales.

Git branch <rama>: Crea una nueva rama.

Git checkout <rama>: cambia el puntero HEAD a <rama>

Git branch –d <rama>: borra la rama.

Git merge <rama>: fusiona el contenido de <rama> en la rama en la que estamos parados.

Deshacer cambios.

Git commit –amend –m <mensaje>: sobreescribe el último mensaje realizado.

Git checkout -- <archivo>: deshace los cambios realizados sobre un archivo.

Git reset: permite volver a versiones anteriores.

Git revert<hash>: revierte el estado sin perder el historial de cambios.

Git stash: descarta los cambios pero guarda una copia de los archivos modificados.