**COMANDOS DE GIT**

1. Comando establecer usuario: git config --global user.name “username”
2. Comando establecer correo: git config --global user.email “correo”
3. Crear un repositorio: Dentro de la carpeta raíz git init
4. Cambiar nombre de la rama principal: git config --global init.defaultBranch name
5. Ver carpetas del repositorio, estado, cambios, etc: git status
6. Dar seguimiento a todos los archivos del repositorio: git add .
7. Quitar seguimiento a un archivo del repositorio: git reset filename
8. Guardar una versión del repositorio: git commit – m “name”
9. Reconstruir el proyecto al ultimo commit: git checkout --
10. Cambiar rama master a main: git branch -m master main
11. Si ya se le da seguimiento a un archivo podemos ahorrar el comando 6 sustiyendo por: git commit -am "name"
12. Ver logs de commit realizados: git log
13. Agregar al stage solo archivos de cierto formato: git add \*.format
14. Agregar todos los tipos de archivos dentro de una carpeta con subdirectorios: git add format/
15. Resumen de datos al log: git status –short
16. Crear un alias a un comando: git config --global alias.tuAlias “comando que sustituye”
17. Log de commits chiquito: git log --oneline
18. Renombrar un commit ya hecho: git commit --amend -m "newName"
19. Para modificar un commit podemos quitarlo del stage: git reset --soft hashDelCommit y después se crea otro commit
20. Regresar a un commit guardando los datos modificados en los archivos: git reset --mixed hashDelCommit
21. Regresar a un commit destruyendo todos los cambios en los archivos: git reset --hard hashCommit
22. Ver un historial de todos los commits guardados: git reflog
23. Cambiar nombre de un commit usando mv: git mv nombre.md nuevo-nombre.md

**CREACION DE RAMAS**

1. Para crear una nueva rama: git Branch nombre
2. Sabe en que rama estas: git Branch
3. Moverte a otra rama: git checkout nombre-rama
4. Para unir una rama a otra hay que estar posicionado en la rama que recibirá la información y hacer: git merge nombre-rama
5. Borrar una rama: git Branch -d nombre-rama
6. Borrar una rama de manera forzada: git Branch -d nombre-rama -f
7. Crear y moverte a una rama de manera automática: git checkout -b nombre-rama

**CREACION DE TAGS**

1. Crear un tag: git tag nombre-tag
2. Eliminar un tag: git tag -d nombre-tag
3. Ver tags existentes: git tag
4. Creación de un reléase tag: git tag -a tag-name -m "mensaje"
5. Crear un reléase tag especifico a un commit: git tag -a tag-name commit-hash -m "mensaje"
6. Ver información de un tag en especifico: git show tag-name

**STASH**

1. Crear stash: git stash
2. Ver los stash: git stash list
3. Recuperar datos del stash: git stash pop
4. Regresar a un stash determinado: git stash apply hashStash
5. Eliminar un stash determinado: git stash dropp hashStash
6. Ver información de un stash: git stash show hashStash
7. Crear un stash con un nombre: git stash save “StashName”
8. Mas información de cada stash: git stash list - -stat

**REBASE**

1. Para hacer una unión de otra rama al master: git rebase master
2. Unir dos commit que deberían ser los mismos: git rebase -i HEAD~numCommit
3. Si tienes múltiples archivos modificados y solo quieres deshacer una: git checkout –name