**Directorios y Ficheros (Crear repositorio)**

echo "Contenido del archivo" > archivo.txt —> Añade texto y lo crea

touch archivo.txt —> Crea documento

mkdir carpeta—> Crea directorio

git init → iniciar repositorio

**Validar repositorios**

Git status—> Estado actual del repositorio

Git add . —> Agregar todos los archivos modificados

Git status—> Estado actual del repositorio

Git comió -m “Comentario”—> Se utiliza en Git para confirmar los cambios realizados en los archivos (HEAD, apunta al último commit)

**Versiones creadas**

git log --oneline —> Muestra el historial de commits de manera resumida

git log → mas extenso

git show → ver cambio en todos los commits

git show “idCommit”--> commit concreto

**Comparar cambios**

git diff → compara el área de working y el comit que llamamos HEAD.

git diff HEAD-1 HEAD → diferencia entre varios commit indicando los apuntadores

git diff id HEAD → consultar la diferencia entre varios commits

**Repositorio remoto**

1. git remote add origin <http://github.com/GemaBT/Hola-Mundo.git>
2. git push
3. git remote -v

**Crear copia con checkout**

1. git init
2. git remote add origin <nombre-repositorio-remoto>
3. git fetch origin master
4. git checkout origin/master

**Revertir resetear commits**

git reset --hard HEAD~1—>resetea el estado de tu directorio de trabajo y el área de preparación al estado del último commit de la rama en la que te encuentras

Git revert HEAD—> Para revertir el último commit en Git

**Conectar con Git Hub**

Git remote add origin<https://github.com/lugosi92/practica_evaluable.git>

Git push -u origin master

git push origin master—> Sincornizar con el repositorio remoto

git push -u origin master→ La ramam como local actual

git pull origin main→ Actualizar rama

**Crear rama y cambiar a ella**

Git branch —> consultar ramas

Git branch mirama

Git checkout mirama→ puede ser con directorios, ficheros etc..

Git checkout -b —> para cambiara ella directamente

//Validar

Git merge mirama—> fusionar ramas

git branch -d mirama —> BORRAR RAMAS

**Crear etiqueta**

**git show <nombretiqueta>**

Git tag -a V1 -m “Etiqueta V1”—> crea una etiqueta anotada llamada "V1"

Git show V1—> Esto mostrará los detalles asociados con la etiqueta "V1",

git tag—> etiquetas existentes

**Clonar Repositorio**

Git clone<https://github.com/lugosi92/practica_evaluable.git> practica\_evaluable\_2

**Provocar conflicto**

1. **Clonar Repositoiro**

git clone < url repositorio remoto>

1. **Cambiar a rama 2**

git branch pruebas2

git checkout pruebas

1. **Crear LÍNEA 1 en archivo creado**

• echo "Creo archivo en pruebas2" > archivo.txt

**3.1 Crear LÍNEA 2**

• echo "Añado otro texto" >> archivo.txt

1. **VALIDAMOS (Añadir a área de trabajo)**

git add archivo.txt

**4.1 Agregar comentario**

• git commit -m "Creando archivo.txt en la rama pruebas2 "

**4.2 Ver histórico ( ver si se ha creado )**

• git log --oneline

1. **Cambiar a MASTER**

git checkout master

1. **Creamos un archivo con el mismo nombre**

• git checkout master

1. **Creamos Líneas nuevas**

• echo "Creo archivo.txt en master" > archivo.txt

1. **Validamos**

• git commit -m "Creado archivo.txt en master"

• git log --oneline

1. **git merge mirama (desde master)**

• En master fusiono, aparece el conflicto. Dos ramas con el mismo nombre indica que hay conflicto al añadir el archivo.txt.

**RESOLVER**

Abrir archivo.txt—> borrar las líneas que no queremos

1. **Validar**

**Sincronizar Repositorio**

git push origin main

**Etiquetas**

git tag -a nombre\_de\_etiqueta -m “Mensaje de la etiqueta”;

git tg -d <nombre\_de\_la\_etiqueta>

git show -nombre\_etiqueta-

git push origin –tags → manualmente al repositorio