

## COMANDOS GIT

Trabajaremos con comandos en la herramienta `git bash`

- Configurar git: (si no lo hacemos no nos dejará hacer commits)
  - `git config --global user.name 'Tu nombre'`
  - `git config --global useer.email tucorreo@alu.murciaeduca.es`
- Iniciar repositorio
  - si estamos dentro de la carpeta: `git init`
  - crea una carpeta con el nombre 'proyecto': `git init proyecto`  
En este caso deberemos entrar después a la carpeta: `cd proyecto`
- Comprobar el estado del working directory: `git status`
- Añadir ficheros al staging area:
  - `git add .` (añade todos los ficheros)
  - `git add nombre-fichero`
- Volver a la versión anterior de un fichero modificado tras hacer commit: `git checkout – fichero`
- Descartar cambios en un fichero (antes de hacer commit): `git restore fichero`
- Sacar un fichero del staging area: `git restore –staged fichero`
- Consultar los ficheros modificados y no preparados (no añadidos al staging area): `git diff`
- Consultar los ficheros modificados y preparados (añadidos al staging area): `git diff -- staged`
- Confirmar cambios (añadir una nueva versión al repo): `git commit –m "Texto descriptivo"`
- Listar los commits realizados: (salir con q)
  - Mostrar todos en detalle: `git log`
  - Mostrar los n últimos en detalle: `git log –n`
  - Mostrar todos incluyendo las modificaciones guardadas: `git log –p`
  - Mostrar todos sin detalle: `git log --pretty=oneline`
- Añadir cambios al último commit: `git commit --amend –m "Nuevo texto"`
- Volver al estado de los fichero de una versión anterior (de un determinado commit): `git checkout hash`
- Etiquetar un commit (permite referenciarlo por la etiqueta en lugar del hash): `git tag etiqueta hash`
- Listar las etiquetas: `git tag –l`
- Crear una rama: `git branch nombre-rama`
- Mover el puntero a una rama: `git checkout nombre-rama`
  - Crear una rama y moverse a ella: `git checkout –b nombe-rama`
- Listar ramas: `git branch`
- Para fusionar ramas:
  - Hacer commit en la rama actual
  - Moverse a la rama main: `git checkout main`
  - Mezclar o fusionar con la rama anterior: `git merge nombre-rama`
- Borrar una rama: `git branch –d nombre-rama`
- Ver la última confirmación de cada rama: `git branch –v`
- Listar ramas (no borradas) fusionadas con la rama main: `git branch --merged`
- Listar ramas (no borradas) pendientes de fusionar con la rama main: `git branch --no-merged`
- Renombrar una rama: `git branch –m nombrenuevo`
  - Renombrar la rama de master a main: `git branch –m main`

## UT06. Documentación y control de versiones

## • TRABAJAR CON REPOSITORIOS REMOTOS

Los repositorios remotos están alojados en github.com

Es conveniente renombrar la rama principal de los repositorios locales de master a main antes de conectar con un repo remoto.

- CLONAR UN REPOSITORIO REMOTO (ya existe en github y tiene contenido que quiero clonar en mi equipo local): `git clone https://URL-repositorio`
  - Traer nuevos cambios del repo remoto al local: `git pull origin main` (donde origin es la rama remota y main o master es la rama local)
- CREAR UN REPOSITORIO REMOTO A PARTIR DE UN REPO LOCAL:
  - Creamos el repositorio remoto en GitHub (estará vacío)
  - Conectar el repo local con el remoto: `git remote add origin https://url-repo-remote`
  - Subir ficheros al repo remoto: `git push -u origin main` (Nos pedirá identificarnos)
- Consultar con qué repo remoto está conectado el repo local:
  - `git remote` (esto nos dirá como se denomina la rama remota)
  - `git remote -v`