

# PRÁCTICA 1 PAT. PRUEBA DE COMANDOS GIT

Enrique Sanz Tur

## - Git clone:

```
cd p1@Enriquesanzzz → /workspaces $ git clone https://github.com/Enriquesanzzz/p1.git
Cloning into 'p1'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

Este comando copia el repositorio que indica la URL que va a continuación (en este caso el repositorio “p1” que he creado desde mi usuario: <https://github.com/Enriquesanzzz/p1.git>). Los logs que aparecen cuando ejecutamos el comando indican que la copia del repo ha sido exitosa.

Viendo la respuesta de la terminal, se ve como el servidor remoto enumera y cuenta los objetos que se encuentran en el repositorio para luego comprimirlos. Después, se muestran una serie de estadísticas, para acabar indicando que todos los objetos del repositorio se han copiado en la maquina local.

## - Git status:

```
@Enriquesanzzz → /workspaces $ cd /workspaces/p1 # Asegúrate de estar en el directorio del repositorio
@Enriquesanzzz → /workspaces/p1 (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Este comando se utiliza para mostrar el estado actual del repositorio. Proporciona información sobre la rama en la que te encuentras, la relación entre tu rama local y la rama remota, y el estado de los archivos en el directorio de trabajo.

En el caso de mi repositorio, viendo los logs de la terminal se puede observar:

- *On branch main* indica que se encuentra en la rama llamada "main", que es la rama predeterminada en muchos repositorios.
- *Your branch is up to date with 'origin/main'.* muestra que la rama local está actualizada con respecto a la rama remota llamada "main" en el repositorio remoto (en este caso, "origin").
- *nothing to commit, working tree clean* indica que no hay cambios pendientes que deban ser confirmados.

- Git add y git commit:

```
● @Enriquesanzzz → /workspaces/p1 (main) $ git add .
⊗ @Enriquesanzzz → /workspaces/p1 (main) $ git commit -m "feat: homepage"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Este comando se utiliza para agregar cambios en el directorio de trabajo a la *staging area* (área de índice). Esta acción prepara los cambios seleccionados para ser incluidos en el próximo commit. Es un paso necesario antes de confirmar los cambios de manera permanente en el historial del repositorio.

Se puede poner *git add <nombre del archivo>* para añadir un archivo concreto, o *git add .* para agregar todos los archivos modificados y nuevos en el directorio de trabajo y sus subdirectorios a la zona de preparación en Git. En otras palabras, prepara todos los cambios para el próximo commit.

Cabe destacar que, después de agregar los cambios con *git add*, es necesario realizar un *commit* para guardar esos cambios en el historial del repositorio. El comando *git commit* se utiliza con el fin de confirmar los cambios en la zona de preparación, con un mensaje, en mi caso *homepage*, para indicar el main como página principal.

En este caso, los logs indican lo mismo que con el comando anterior:

- *On branch main* indica que se encuentra en la rama llamada "main", que es la rama predeterminada en muchos repositorios.
- *Your branch is up to date with 'origin/main'.* muestra que la rama local está actualizada con respecto a la rama remota llamada "main" en el repositorio remoto (en este caso, "origin").
- *nothing to commit, working tree clean* indica que no hay cambios pendientes que deban ser confirmados. Básicamente, que el directorio de trabajo está limpio.

- Git push:

```
⊗ @Enriquesanzzz → /workspaces/p1 (main) $ git push origin main
remote: Permission to Enriquesanzzz/p1.git denied to Enriquesanzzz.
fatal: unable to access 'https://github.com/Enriquesanzzz/p1.git/': The requested URL returned error: 403
⊗ @Enriquesanzzz → /workspaces/p1 (main) $ git push origin main
remote: Permission to Enriquesanzzz/p1.git denied to Enriquesanzzz.
fatal: unable to access 'https://github.com/Enriquesanzzz/p1.git/': The requested URL returned error: 403
● @Enriquesanzzz → /workspaces/p1 (main) $ git remote -v
origin https://github.com/Enriquesanzzz/p1.git (fetch)
origin https://github.com/Enriquesanzzz/p1.git (push)
● @Enriquesanzzz → /workspaces/p1 (main) $ git remote set-url origin git@github.com:Enriquesanzzz/p1.git
⊗ @Enriquesanzzz → /workspaces/p1 (main) $ git push origin main
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOtrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,140.82.121.3' (ECDSA) to the list of known hosts.
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

Este comando se utiliza en Git para enviar tus cambios locales en la rama "main" a la rama correspondiente en el repositorio remoto llamado "origin". es útil para mantener sincronizadas las versiones locales y remotas de tu código, permitiendo que otros colaboradores vean y obtengan esos cambios.

En mi caso, hay un problema con los permisos, que están denegados. En la configuración del repositorio en GitHub, he intentado ponerme como colaborador sin éxito, ya que no se puede ser creador y colaborador a la vez. También revisar todas las políticas de privacidad y seguridad para asegurarme de que tengo todos los permisos, comprobando que todo está en orden. Sin embargo, al seguir intentando lanzar el comando, seguía saliendo el mismo problema.

En otro intento de seguir arreglándolo, he escrito el comando `git remote -v` para verificar la configuración remota del repo. La salida indica que la URL del repositorio es de tipo HTTPS y apunta a tu repositorio en GitHub.

En otro intento, he intentado cambiar la URL remota a la versión SSH con el siguiente comando, con el objetivo de no usar HTTPS para la autenticación: `git remote set-url origin git@github.com:Enriquesanzzz/p1.git`. Pero los logs del terminal indican que se encontró un problema con la autenticación mediante la clave SSH. Después de aceptar la autenticidad del host, sale un mensaje de *"Permission denied (publickey)"* indicando que no se pudo acceder al repositorio remoto debido a problemas de permisos.

- Git checkout:

```
● @Enriquesanzzz → /workspaces/p1 (main) $ git checkout -b feat/add-body
  Switched to a new branch 'feat/add-body'
● @Enriquesanzzz → /workspaces/p1 (feat/add-body) $ git add .
  origin feat/add-body

● @Enriquesanzzz → /workspaces/p1 (feat/add-body) $ git checkout main
  Switched to branch 'main'
  Your branch is up to date with 'origin/main'.
```

Este comando se utiliza para realizar varias operaciones relacionadas con el control de versiones, y su comportamiento puede variar según el contexto.

En este caso, con el primer comando `git checkout -b` se crea la rama *feat/add-body* y se cambia a esa rama, tal y como indica la respuesta del terminal.

También, el resultado del comando `git checkout main` indica que has cambiado con éxito a la rama "main" en tu repositorio. Aquí está la interpretación:

- *Switched to branch 'main'* significa que se ha cambiado con éxito a la rama llamada "main".
- *Your branch is up to date with 'origin/main'* indica que la rama local "main" está actualizada con respecto a la rama remota "main" en el repositorio remoto llamado "origin". No hay nuevos cambios en la rama remota que no hayan sido incorporados a la rama local.

