Git notes

Crear rama y hacer tracking con rama remota

**Crear una nueva rama local llamada 'A\_Branch' y cambiar a ella**:

bash

git checkout -b A\_Branch

Este comando crea una nueva rama local llamada **'A\_Branch'** y cambia a ella.

**Empujar la nueva rama local al repositorio remoto**:

bash

git push -u origin A\_Branch

El argumento **-u** establece la rama local recién creada para que rastree a la rama remota con el mismo nombre, lo que significa que en futuros git push y git pull, Git sabrá a qué rama remota enviar o desde dónde traer los cambios.

Luego de ejecutar este comando tendremos una salida similar :

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0

remote:

remote: Create a pull request for 'A\_Branch' on GitHub by visiting:

remote: https://github.com/User/gitRepo/pull/new/A\_Branch

remote:

To github.com:User/MyProject.git

\* [new branch] A\_Branch -> A\_Branch

branch 'A\_Branch' set up to track 'origin/A\_Branch'.

**Traer los cambios de la rama remota por encima de la rama local**

Para descartar los cambios locales en tu rama A\_Branch y volver al estado del último commit registrado en la rama remota A\_BranchRemote, puedes seguir estos pasos:

1. Asegúrate de estar en la rama A\_Branch:

bash

git checkout A\_Branch

1. Descarga los cambios más recientes de la rama remota A\_BranchRemote:

bash

git fetch origin A\_BranchRemote

1. Reemplaza los cambios locales con los de la rama remota A\_BranchRemote. Puedes hacerlo de varias maneras, pero una forma segura es hacer un reset duro:

bash

git reset --hard origin/A\_BranchRemote

Con estos pasos, habrás descartado los cambios locales en la rama A\_Branch y habrás actualizado tu rama local con los cambios más recientes de la rama remota A\_BranchRemote. Asegúrate de que no tengas cambios importantes en la rama A\_Branch que desees conservar, ya que el reset duro eliminará todos los cambios locales no confirmados.

**Si es que persisten archivos al hacer un git status**

Si realizaste un reset duro con git reset --hard origin/A\_BranchRemote y aún así tienes archivos no rastreados, es posible que estos archivos no estén bajo control de versiones en tu repositorio remoto y, por lo tanto, no fueron afectados por el reset duro.

Para eliminar los archivos no rastreados, puedes ejecutar el siguiente comando:

bash

git clean -fd

Este comando eliminará todos los archivos no rastreados en tu directorio de trabajo. Asegúrate de revisar los archivos no rastreados antes de ejecutar este comando, ya que eliminará los archivos de forma permanente. Si hay archivos que deseas conservar, muévelos a un directorio seguro antes de ejecutar git clean -fd.

**Subir un Repositorio local a uno remoto (con ssh)**

Para este ejemplo usaremos el repositorio local llamado ‘veterinary’ y el repositorio remoto

<https://github.com/demargin/veterinary>

git@github.com:demargin/veterinary.git

1. Primero crear un repositorio remoto en github
2. Conectar el repositorio Local a Github
   1. git remote add origin <https://github.com/demargin/veterinary>
   2. Verificar con el siguiente comando :
      1. Git remote –v
3. Subir los cambios con “git add .” y con “git commit –m “Mensaje” “
4. Ver las claves instaladas en el ordenador con :
5. ls –al ~/.ssh  
     
   Resultado de ejemplo  
     
   -rw------- 1 devmartin devmartin 419 Jan 24 22:20 id\_ed25519   
   -rw------- 1 devmartin devmartin 419 Mar 11 16:17 id\_ed25519\_devMartin   
   -rw-r--r-- 1 devmartin devmartin 108 Mar 11 16:17 id\_ed25519\_devMartin.pub   
   -rw-r--r-- 1 devmartin devmartin 104 Jan 24 22:20 id\_ed25519.pub

Crear o modificar el archivo ‘config’ de SSH  
nano ~/.ssh/config   
  
Ejemplo:  
  
# Configuración para GitHub con clave personal

Host github.com-personal

HostName github.com

User git

IdentityFile ~/.ssh/id\_ed25519

# Configuración para GitHub con clave de trabajo

Host github.com-trabajo

HostName github.com

User git

IdentityFile ~/.ssh/id\_ed25519\_devMartin

### 5. Asociar la configuracion con la URL del Repositorio: Ejemplo: git remote set-url origin [git@github.com-personal:demargin/veterinary.git](mailto:git@github.com-personal:demargin/veterinary.git) Verificar la configuracion: git remote -v 6. Realizar push al repositorio remoto git push origin main \*\*\*Nota puede que haya que resolver conflictos: **Rebase tus Cambios sobre la Rama Remota**

Si prefieres reescribir la historia de tu rama local sobre la base de la rama remota, puedes usar el rebase. Esto aplicará tus commits locales sobre la rama remota, uno por uno. Ejecuta el siguiente comando:

bash

git pull origin main --rebase

Git realizará un rebase de tu rama local (main) sobre la rama remota (origin/main). Si hay conflictos, Git te pedirá que los resuelvas durante el proceso de rebase.  
  
si aparecen mas conflictos se debe verificar con **git status** que archivos estan causando los conflictos, luego con **git add .**  Agregar los archivos conflictivos y usar **git rebase —continue** , finalmente si ya no existe ningun conflicto, empujar los cambios del repo local hacia la rama remota con  **git push origin main .**