

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

Programacion 2 - Miercoles

CLASE 6

GRUPO SOLUCION 202

INTEGRANTES:

- ORELLANA JUAN MARCOS
- VELAZQUEZ OMAR IGNACIO
- AULAR AFRICA
- GAUNA GIMENA
- SANHUEZA MARIANELA FERNANDA
- HERRERA EMMANUEL

ACTIVIDADES

Error con los tags

Investigación: ¿Qué pasa si por error cargamos un tag dos veces? ¿Cómo solucionarías este problema o error?

¿Qué pasa si creamos un tag dos veces?

Si el tag ya existe en el repositorio remoto y lo intentamos crear de nuevo en otro commit (o el mismo), Git mostrará el siguiente error:

- error: tag <'nombre_del_tag'> already exists

Esto sucede porque Git no permite sobrescribir un tag ya existente a menos que se fuerce.

¿Cómo solucionarías este problema o error?

Existen varias formas de abordar esta situación, dependiendo del objetivo:

1. Eliminar el tag duplicado:

Identificar el tag:

- git tag

Este comando mostrará todos los tags existentes en tu repositorio.

Eliminar el tag:

- `git tag -d <nombre_del_tag>`

Eliminar el tag en el repositorio remoto:

Si ya se empujó el tag al repositorio remoto, es necesario eliminarlo del remoto también. Esto se hace con el siguiente comando:

- `git push origin --delete <nombre_del_tag>`

2. Renombrar el tag duplicado:

Renombrar el tag:

- `git tag -m "Nuevo mensaje" <nuevo_nombre> <nombre_del_tag>`

3. Ver el historial de un tag:

Si deseamos ver a qué commit apuntaba un tag en el pasado, podemos utilizar:

- `git log -1 --pretty='%H' <nombre_del_tag>`

Este comando mostrará el hash del commit al que apuntaba el tag en un momento dado.

4. ¿Qué hacer si el tag ya fue "pushed" y queremos actualizarlo?

Si se prefiere forzar la actualización del tag remoto sin eliminarlo explícitamente (por ejemplo, si varias personas están usando ese tag), se puede usar el siguiente comando:

- `git push --force origin <nombre_del_tag>`

Este comando sobrescribe el tag en el remoto con el commit en el que se ha recreado localmente.

¿Por qué usamos estos comandos?

- **git tag:** Este comando es fundamental para gestionar tags en Git. Se utiliza para crear, listar y eliminar tags.
- **-d:** El argumento -d se utiliza para eliminar un tag.
- **-m:** El argumento -m se utiliza para modificar el mensaje de un tag al renombrarlo.
- **git log:** Este comando se utiliza para ver el historial de commits. Los argumentos -1 y --pretty='%H' se usan para mostrar únicamente el hash del commit más reciente del tag.

