1 Spiegare ruolo di “origin”

2 Comandi per creazione repository da riga di comando

3 Eseguire una git push force e spiegare le conseguenze di tale azione

4 Ogni commit ha un id univoco. Documentatevi sul come tale id viene assegnato. (non è necessario scendere troppo nel dettaglio).

5 Spiegare differenza fra una revert e una reset, con relativi esempi

6 Spiegare i due metodi di integrazione branch visti finora, con relativi esempi

1.Il termine origin lo troviamo nel comando push –u origin master oppure push -u origin develop che sono i comandi utlizzati per creare un nuovo branch(sia esso master o develop) in una repository locale caricandola(upload) su una repository remota indicata col termine “origin”.

2.Esistono due modi per creare una repository da linea di comando:

1. Il primo modo prevede il comando **$ git init [project-name]** che crea una repository locale con un nome specifico in una directory esistente, a seguito di tale comando abbiamo bisogno di aggiungere l’url di una reposity remota alla nostra configurazione locale (Creata) e lo facciamo con il comando “add” , poi creiamo il nostro branch master su remoto(il termine origin), queste due operazioni le eseguiamo rispettivamente con i comandi

**$ git remote add origin <remote\_repo\_url>**

**$** **git push -u origin master**

1. Il secondo modo prevede di clonare una requsitory già esistente con il comando

**$ git clone [url]**

Che carica tutta la sua storia cronologia.

3.Il comando **di git push[nome] - force o git push -f** cancella la commit più recente ed effettua un push sulla commit attuale per cui se si effettua una merge la force sovrascrive la commit precedente.

4. Ogni commit è identificato con un id univoco che è una stringa in codice SHA-1 di tutte le info più importanti sulla commit tra cui:

la data del commit,il nome e l’indirizzo e-mail del committer ,il messaggio di log(comanado  **git commit –m “messaggio”**, e l’id del precedente commit.

5. I comandi di reverse e reset servono ad annullare le modifiche effettuate ad un file ma con delle differenze.

Con il comando **$ git fetch origin** visualizziamo tutti i cambiamenti e i commit locali fatti e con il comando

**git reset --hard origin/master** annulliamo il branch master locale, con il comando **$ git reset HEAD <file>** annulliamo tutte le modifiche fatte ai file che si trovano nella head,

con il comando **$ git revert**

annulliamo tutti i cambiamenti fatti ad una singola commit senza perdere le modifiche fatte ai commit successivi, quando si lancia questo comando si apre una nuova commit in cui ci viene richiesto di inserire un messaggio, il message del commit altrimenti possiamo inserire **$ git revert<commit>**  dove commit è il codice ash della commit di interesse .

6. I due metodi di integrazione del branch da un ramo all’altro sono il merge e il rebase che si eseguono rispettivamente con i comandi:

**$ git merge --no-ff [branch name]** effettua un merge nel branch corrente;

**$ git merge origin/develop** effettua un merge del branch develop nel branch corrente;

Il rebase invece si esegue con il comando **git rebase <base>** dove  **base** possono essere l’id di un commit, il nome di un beanch, un tag o un riferimento relativo a HEAD. Con il merge perdiamo la storia delle precedenti commit e ne creiamo una di merge, il rebase non fa altro che prendere tutti i cambiamenti che sono stati applicati ad un ramo e applicarli su un altro.