**1.Spiegare ruolo di “origin”.**

Origin fa riferimento all’indirizzo del repository in remoto.

**2.Prendere dimestichezza con i comandi visti.**

**3.Comandi per creazione repository da riga di comando.**

Il repository è la location dove sono inseriti i file relativi ad un determinato progetto. Si può inizializzare o clonare.

Inizializzare il repository significa tracciare un progetto partendo da zero. Il comando $ git init [project-name] crea un repository locale con il nome specificato in una directory già esistente ($ mkdir dir\_repo && cd dir\_repo). Dopo si dovrà associare un riferimento a un repository remoto alla configurazione locale, e settare un branch principale per i branches locali con i comandi:

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

$ git push -u origin master

Clonare un repository significa scaricare un progetto già tracciato che si trova online, e lo si fa con il comando $ git clone [url].

**4.Eseguire una git push force e spiegare le conseguenze di tale azione.**

Normalmente il comando push rifiuta di aggiornare un riferimento remoto che non è un antenato del referente locale utilizzato per sovrascriverlo. Inoltre, quando viene utilizzata l'opzione force-with-lease, il comando rifiuta di aggiornare un riferimento remoto il cui valore corrente non corrisponde a quello previsto.

L’aggiunta del flag force disabilita questi controlli ma può causare la perdita delle commit del repository remoto, quindi si deve utilizzare con attenzione.

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

Il codice Id è un codice hash SHA-1, un codice di 40 cifre in esadecimale, ottenuto su informazioni come il messaggio di commit, l’autore della commit, il contenuto dei tre alberi.

**6.Spiegare differenza fra una revert e una reset, con relativi esempi.**

Il comando revert annulla una commit errata creando una nuova commit, cioè ricrea la commit prima di averla modificata.

Il comando reset toglie la commit e cancella i file presenti nella stage area.

$ git reset HEAD <file>

**7.Spiegare i due metodi di integrazione branch visti finora, con relativi esempi.**

La merge effettua l’unione dei due branch però non tiene traccia della storia delle commit. La rebase compie la stessa procedura ma tiene traccia della storia delle due commit che fonde.