# HomeWork Day 2

**Siete il release manager di un progetto, descrivete il git flow che tutto il team dovrà seguire.**

Partendo dall’inizializzazione del repository,tutto il team farà riferimento a tale repo con il nome di “Origin”. Ogni elemento del team “pusherà” e “pullerà” verso l’origin, in particolare, con il primo si inviano i dati in remoto mentre con il secondo di aggiorna in locale nel caso in cui siano state apportate modifiche da altri.

In particolare dal master(progetto originale) sarà clonato un ramo “develop” su cui andare ad apportare modifiche e quindi aggiornamenti. Ogni parte del team avrà diverse funzionalità da eseguire quindi task, che si svolgeranno su diversi branch staccati dal ramo develop.

**Spiegate le naming convention che volete adottare, le modalità di commit e integrazione dei vari branch.**

Eseguendo una naming convention classica, ci si riferirà ai FEATURES per delle nuove funzionalità da eseguire, che possono essere IMPROVE per dei miglioramenti opp BUG ossia errori da risolvere. Altri tipi di branch a cui ci si riferirà saranno RELEASE per il rilascio del progetto sia sul develop che sul master ed a branch HOTFIX errori molto gravi che devono essere risolti nell’immediato, per tale motivo solo in qst caso si stacca il branch dal master.

I branch Futeres saranno staccati dal develop risolti da ogni team in working space. Le modifiche effettuate saranno “aggiunte” all’area stage per poi essere successivamente commitate nell’head. Prima di pushare le modifiche all’interno del develop è buon uso “mergiare” il develop sul proprio branch per risolvere eventuali conflitti che possono incorrere. Una volta usciti poi dall’area di merge siamo pronti a pushare in remoto.

Per i branch Hotfix e Release avviene lo stesso procedimento a meno del merge finale che avverrà sia sul master che sul develop, questo per tenere traccia delle modifiche apportate per i successivi branch che vorremmo staccare dal develop.

**In caso di hotfix, descrivete la modalità con cui volete procedere per fixare eventuali bug nel miglior modo possibile e come integrare tali fix.**

Come detto sopra, in caso di Hotfix, sarà staccato tale branch dal master, risolto il prima possibile, per poi essere mergiato sia sul ramo master che quello develop. Lo si fa anche sul secondo per aggiornare la versione del master sul develop in modo che vengano apportate immediatamente le modifiche sulle parti del progetto che riportavano problemi.

**Descrivete inoltre tutte le operazioni che non devono essere eseguite per preservare l’integrità del progetto.**

Per preservare le integrità del progetto ogni branch non deve mai essere staccato dal master, a meno di develop e Hotfix ed non bisogna forzare una push perché potrebbe importare problemi nell’intero progetto tranne nel caso in cui si è completamente convinti di ciò che si fa.

**Spiegare cosa è una Pull Request**

La “pull request” è l’avviso che viene fatto ai collaboratori della repository, dopo aver eseguito la push di un lavoro sul remoto. Lo si può fare tramite il sito *GitHub* dal pulsante "pull request" che automaticamente notifica i collaboratori oppure eseguire il comando “git request-pull” e inviare manualmente l'output via email.