# HomeWork Day 2

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

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

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

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

Come esercitazione:

Staccare un branch dal develop del vostro repository (CognomeNome)

Nelle cartella HomeWork pushare il vostro homeWork2.

Spiegare cosa è una Pull Request e fare la richiesta da GitHub indicando “uiip32” come reviewer.

Il git flow che il team dovra seguire consiste in due rami principali:

1. **Il branch master** che memorizza la storia della release ufficiale. Rappresenta l’immagine del progetto in produzione e ogni volta che viene rilasciata una release è necessario taggare tutti i commit nel ramo principale con il numero di versione.
2. **Il branch develop** che serve come ramo di integrazione per le features.

Ogni componente del team potrà clonare il progetto dal repository remoto e attuare le modifiche nel repository locale per poi pusharle in remoto.

*Per ogni feature verrà creato un nuovo branch dal develop* e quando lo sviluppo della feature sarà completa essa *verrà mergiata nel branch padre, cioè nel develop*.

*Le features vengono staccata dal ramo develop e mai dal master, inoltre non devono mai interagire con il master.*

*Se ci sono dei conflitti quando si effettua il push delle commit nel repository remoto, questi vanno risolti e poi si effettua l’integrazione tramite merge o rebase. E’ buona prassi far risolvere i conflitti all’utente che ha lavorato su quella funzionalità.*

Come name convention usiamo per ogni nuova Feature staccata dal develop :

-Feature -> quando sviluppiamo una nuova funzionalità

-Improve -> quando apportiamo un miglioramente ad una funzionalità già esistente

-bug -> quando c’è un bug che deve essere risolto.

Quando abbiamo sviluppato un numero tale di Feature o si è a ridosso di una scadenza bisogna creare un **branch Release** che viene staccato dal develop. Qui inizia la fase di rilascio dove vengono effettuati i test necessari e fissati i bug. Da questo momento non possono essere più aggiunte Feature alla release.

Quando la Release è pronta il branch di Release viene mergiato nel master con la relativa etichetta di versionamento *e mergiato anche nel develop che può trovarsi in uno stato avanzato rispetto a quando il ramo di release è stato staccato.*

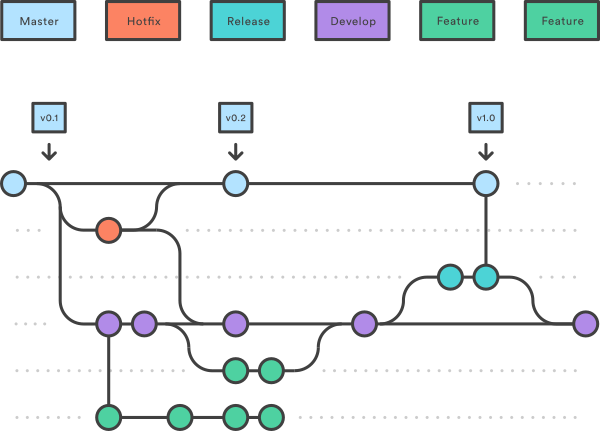
Usiamo anche un ulteriore **branch** denominato **HotFix** per correggere rapidament i bug gravi presenti nella release in produzione.

*Se alcune Feature sono strettamente collegate alla funzionalità che si sta correggiendo nel branch HotFix è meglio sospendere il lavoro su queste Feature e riprenderlo dopo aver fissato il bug.*

*Questo branch è l’unico che si stacca direttamente dal master* e ci consente di affrontare il problema senza interrompere il flusso di lavoro o attendere la release successiva.

Per questo motivo una volta completata la correzione si effettua *il merge di questo branch* sia nel master con la relativa etichetta di version update*, sia nel branch develop.*

*\*in corsivo le operazioni per preservare l’integrita del progetto.*



Una Pull Request è un meccanismo usato dal programmatore per informare i membri del team di avere completato una Feature.

La Pull Request è più di una notifica: è un forum dedicato per discutere della funzionalità proposta. Se ci sono problemi con le modifiche, i membri del team possono inviare feedback nella Pull Request e persino modificare la Feature effettuando il push delle commits seguite.