

Risoluzione di un bug

Sei uno sviluppatore ed è stato trovato un bug in una classe Java sviluppata da un tuo collega. Per policy aziendale, per risolvere un bug bisogna prima creare un branch che si chiama *hotfix_n* dove al posto di n va il numero del bug trovato fino a quel momento dai tester. Risolto il bug si può richiedere di riportare la modifica sul progetto principale

Più in dettaglio, i passi da effettuare sono:

1. Creare una *fork* del progetto <https://github.com/carmelo-cina-sal/Operazioni> sul tuo account GitHub. Se l'operazione è stata fatta correttamente, dovrai trovarti il progetto sul tuo repository.
2. Effettuare la *git clone* della nuova repository sul tuo PC locale, sotto *c:\Utenti\studente\PortableGit\repository*.
3. Spostarsi sotto la directory del nuovo progetto appena clonato.
4. Tramite *git branch* creare un *branch* chiamato *hotfix_32*.
5. Spostarsi sotto il branch creato (*git checkout*).
6. Con un editor di testo, correggere la riga 23 del file Operazioni.java, sostituendo il meno (-) con il più (+).
7. Per poter aggiungere tali modifiche al repository, sarà necessario utilizzare il comando *git add*.
8. Al fine di allineare il tuo repository remoto, effettua la *commit* sul tuo repository locale e infine effettua la *push* in remoto
9. Su GitHub, dalla copia del tuo repository, effettua la Pull request al progetto principale (ricorda di allineare i due repository a partire dal tuo branch). **Nel titolo della Pull request metti il tuo Nome e Cognome, nel commento incolla i comandi inseriti sulla bash di git.** Sei non sei arrivato a fare la pull request, incolla i comandi in un file di testo e invialo a carmelo.cina@iistommasosalvini.edu.it giuseppe.schiavone@iistommasosalvini.edu.it

Di seguito uno schema di riepilogo dei comandi principali di Git.

Configurazione globale <i>Configurazione dell'utente valida per tutti i repository</i> \$ git config --global user.name "[name]" Imposta il nome che vuoi mostrare sulle tue commit \$ git config --global user.email "[email address]" Imposta l'email che vuoi mostrare sulle tue commit	Creare repository <i>Crea un nuovo repository o clonane uno esistente da un URL</i> \$ git init [project-name] Crea un nuovo repository locale con il nome specificato \$ git clone [url] Scarica un progetto esistente e il suo storico di cambiamenti
Effettuare modifiche <i>Rivedi i cambiamenti al codice e prepara una commit</i> \$ git status Elenca tutti i file nuovi o modificati \$ git diff Mostra le differenze non ancora nell'area di staging \$ git add [file] Crea uno snapshot del file in preparazione al versioning \$ git diff --staged Mostra le differenze tra staging e ultima modifica	Rivedere lo storico <i>Esplora l'evoluzione dei file del progetto</i> \$ git log Elenca lo storico di versione per il branch corrente \$ git log --follow [file] Elenca lo storico di versione per il file specificato, incluse rinominazioni \$ git diff [first-branch]...[second-branch] Mostra la differenza tra due branch \$ git show [commit]

<pre>\$ git reset [file]</pre> <p>Rimuovi un file dall'area di staging, ma mantieni le modifiche</p> <pre>\$ git commit -m "[descriptive message]"</pre> <p>Salva gli snapshot dei file in maniera permanente nello storico</p>	<p>Mostra i metadati e i cambiamenti della commit specificata</p>
<p>Annullare commit <i>Elimina errori e altera lo storico dei cambiamenti</i></p> <pre>\$ git reset [commit]</pre> <p>Annulla tutte le commit effettuate dopo [commit], preservando i cambiamenti locali</p> <pre>\$ git reset --hard [commit]</pre> <p>Elimina tutto lo storico e i cambiamenti fino alla commit specificata</p> <pre>\$ git restore [file]</pre> <p>Allinea il file locale al repository</p>	<p>Sincronizzare i cambiamenti Collegati a un URL remoto e ottieni lo storico dei cambiamenti</p> <pre>\$ git fetch [remote]</pre> <p>Scarica lo storico dei cambiamenti dal repository remoto</p> <pre>\$ git merge [remote]/[branch]</pre> <p>Unisci il branch remoto con quello locale</p> <pre>\$ git push --set-upstream [remote] [branch]</pre> <p>Carica tutti i cambiamenti dal branch locale su GitHub</p> <pre>\$ git pull</pre> <p>Scarica lo storico e unisci i cambiamenti</p> <pre>\$ git branch [branch-name]</pre> <p>Crea un nuovo branch</p>