

Procedura di sviluppo

per iniziare:

1. fork da doceo/plant
2. clone in locale del repository remoto

<https://git-scm.com/book/it/v2/GitHub-Maintaining-a-Project>

<https://git-scm.com/book/it/v1/Basi-di-Git-Lavorare-coi-server-remote>

per ogni features da realizzare:

1. aggiornamento all'ultima versione del branch master di doceo/plant (quindi prima bisogna aggiornare il master in locale e poi push del branch master in remoto). Ovviamente per farlo bisogna aggiungere il repository remoto (doceo/plant) <https://gist.github.com/CristinaSolana/1885435> e successivamente caricare gli ultimi aggiornamenti
2. creazione del branch (il nome è legato alla features e può essere indicato nella issue da risolvere)
3. merge del branch features_x con il master in locale.
4. aggiunta delle modifiche all'index locale
5. commit delle modifiche
6. push nel branch master sul repository utente delle modifiche in stato di commit (<https://git-scm.com/book/it/v1/Basi-di-Git-Salvare-le-modifiche-sul-repository>)
7. Pull request verso doceo/plant delle modifiche realizzate. il commento alla pull request deve avere all'interno un riferimento alla issue soddisfatte.

Gestione del repository remoto

```
cd into/cloned/fork-repo
```

```
git remote add upstream
```

```
git://github.com/ORIGINAL-DEV-USERNAME/REPO-YOU-FORKED-FROM.git
```

```
git fetch upstream
```

Manuale per l'utilizzo corretto dei comandi di Git

la guida messa a disposizione da git che utilizziamo come riferimento è:

<https://git-scm.com/book/it/v1/Per-Iniziare-II-Controllo-di-Versione>

1) Fare la pull del branch:

Aprire Git Bash;

Cambiare la directory corrente di lavoro all'interno del progetto locale;

Controllare il ramo a cui ci si vuole unire , solitamente si tratta del master (attraverso il comando "\$ git checkout master");

Tirare il ramo desiderato dal repository upstream, lasciando la cronologia del commit invariata ("\$ git pull <https://github.com/> ORIGINAL_OWNER / ORIGINAL_REPOSITORY .git BRANCH_NAME");

Risolvere eventuali conflitti;

Impegnare il merge

Riesaminare le modifiche per capire se siano soddisfacenti o meno;

Spingere il merge nella propria repository di GitHub ("\$ git push origin master").

2) Creare un branch a partire dal proprio master:

I branch (ramificazioni) sono utilizzati per sviluppare features che sono isolate l'una dall'altra. Il branch master è quello di default quando crei un repository. Puoi usare altri branch per lo sviluppo ed infine incorporarli (merge) nel master branch una volta completati. Per creare un branch a partire dal proprio master si procede nel seguente modo:

Creare un nuovo branch chiamato "feature_x" e passare al nuovo branch usando "git checkout -b feature_x";

Ritornare di nuovo sul master grazie al comando "git checkout master";

Una volta qui, cancellare il branch creato in precedenza con "git branch -d feature_x", in modo da non rendere disponibile il branch agli altri fino a quando non verrà inviato al repository remoto;

Concludere con "git push origin <branch>".

3) Fare un commit delle modifiche con relativo push sul branch del proprio account:

Quando si editano dei file, Git li vede come modificati, perché sono cambiati rispetto all'ultima commit. Bisogna mettere nell'area di stage i file modificati per poi fare la commit di tutto ciò che è in quest'area e, quindi, ci sarà la ripetizione del ciclo.

Successivamente, per inviare le modifiche sul branch del proprio account, bisogna utilizzare il comando "git push origin master" e cambiare master nel branch al quale si vogliono inviare i cambiamenti.

4) Fare una pull request:

Andare sulla pagina principale del repository;

Nel menù "branch", scegliere il branch che contiene i vari commit;

Fare clic su "Nuova pull request", a destra del menù Branch;

Utilizzare il menù a discesa del branch di base per selezionare il branch sul quale unire le modifiche, per poi utilizzare il menù a discesa del branch di confronto per scegliere il branch dell'argomento su cui apportare le modifiche;

Digitare un titolo ed una descrizione per la richiesta di pull;

Cliccare su "Crea nuova pull request".

5) Aggiornare il proprio master:

Per aggiornare il tuo repository locale alla commit più recente, eseguire il comando "git pull" nella directory corrente per fare una fetch (recuperare) ed incorporare (merge) le modifiche fatte sul server remoto.

Per incorporare un altro branch nel branch attivo (ad esempio master), utilizzare "git merge <branch>".

In entrambi i casi git prova ad auto-incorporare le modifiche. Sfortunatamente, a volte questa procedura automatizzata non è possibile, ed in questo caso ci saranno dei conflitti. L'utente stesso è il responsabile che sistemerà questi conflitti manualmente modificando i file che git mostrerà. Dopo aver cambiato questi files, urgerà marcarli come 'correttamente incorporati' tramite "git add <nomedelfile>"; prima di immettere le modifiche, si potrà anche visualizzare un'anteprima eseguendo "git diff <branch_sorgente> <branch_target>".

6) Chiudere un branch:

Il metodo più efficace per chiudere un branch di feature è il seguente: combinare il merge commit con il flag --close-branch (ad esempio, commettere i file modificati e chiudere il branch simultaneamente):

```
hg up feature-x
```

```
hg merge default
```

```
hg ci -m "Merge feature-x and close branch" --close-branch
```

```
hg branch default -f
```

Poi esistono casi particolari di branch chiusi in default.