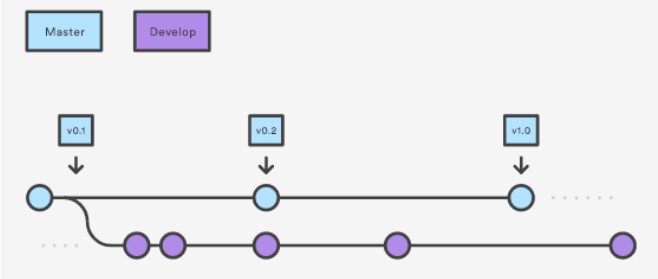
GIT FLOW

(https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow)

Creazione master e develop



Per creare un git flow senza git-flow extension

Si Lancia il git clone

Poi per creare il master:

git branch develop

git push -u origin develop

Oppure con le estensioni git-flow:

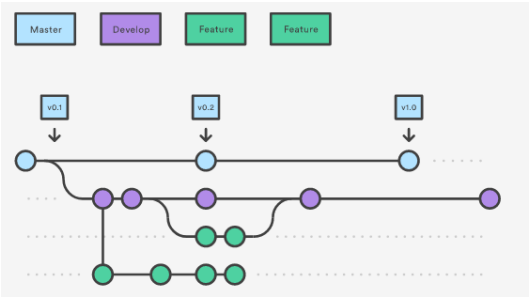
```
$ git flow init
Initialized empty Git repository in ~/project/.git/
No branches exist yet. Base branches must be created r
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []

$ git branch
* develop
  master
```

il branch develop conterrà la storia del progetto mentre il master solo una versione ridotta

Il branch feature.



Viene creato ogni qual volta si sviluppa una nuova caratteristica, generalmente rimane in locale ma può essere pushato nella repository centrale per la collaborazione con altri sviluppatori, **aprire e chiudere molti feature branches, idealmente uno per features**.

I rami delle features costituiscono a tutti gli effetti con il ramo di sviluppo il flusso di lavoro principale e **vengono creati sull'ultimo commit del branch develop**.

Il ramo parente del ramo feature è develop e mai master!.

Per creare un ramo feature:

git checkout develop

git checkout -b feature_branch

ultilizzando le estensioni git-flow:

git flow feature start feature_branch

Per chiudere un ramo feature:

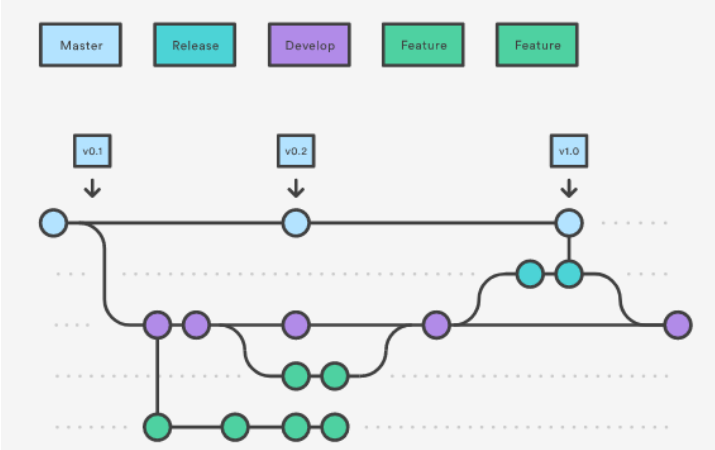
git checkout develop

git merge feature_branch

ultilizzando le estensioni git-flow:

git flow feature finish feature_branch

Il Branch release



Quando si sono accumulate un numero abbastanza cospiquo di features è ora di creare un branch release dal branch develop. **Nessuna nuova features verrà aggiunta a questo punto ma solo fix dei bug**. Quando la **release è pronta per andare online viene mergiata nel branch master e taggata col numero di versione**.

Creare un banch release permette di rilasciare una release mentre

L'uso di un ramo dedicato per preparare la release consente a un team di perfezionare la release corrente correggendo i bachi mentre un altro team continua a lavorare sulle funzionalità per la versione successiva.

Crea un branch release senza le estensioni git-flow

git checkout develop

git checkout -b release/0.1.0

git tag -a v1.0.2 -m "Questa è la versione 1.0.2"

con le estensioni git-flow:

$ git flow release start 0.1.0

Switched to a new branch 'release/0.1.0'

Una volta che la release è pronta va mergiata nel master e nel develop, e quindi la release corrente va cancellata.

Per mergiare nel master la release senza le estensioni git-flow:

git checkout master

git merge release/0.1.0

git checkout develop

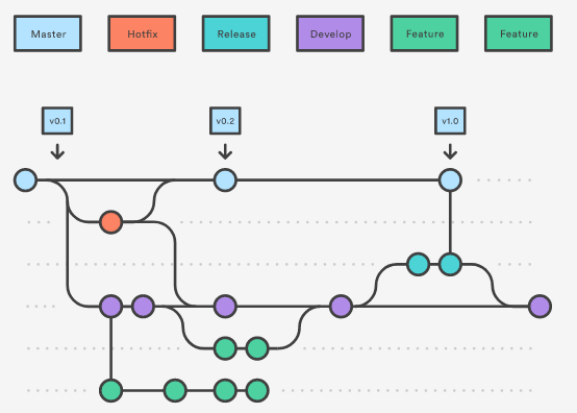
git merge release/0.1.0

delete del branch release…

con le estensioni git-flow:

git flow release finish '0.1.0'

Hot fix branches



Se vi è qualche baco in produzione possiamo fissarlo con l'hot fix, una volta finita la modifica occorre mergiare sul master e sul develop la modifica (o sulla release se è in corso una release). Il master dovrebbe essere taggato con un update version number.

Come creare un branch hotfix senza estensioni git-flow:

git checkout master

git checkout -b hotfix_branch

Con estensioni git-flow

$ git flow hotfix start hotfix_branch

Per chiudere un hotfix:

git checkout master

```
git merge hotfix_branch
git checkout develop
git merge hotfix_branch
git branch -D hotfix_branch
```

con estensioni di git-flow:

```
$ git flow hotfix finish hotfix_branch
```