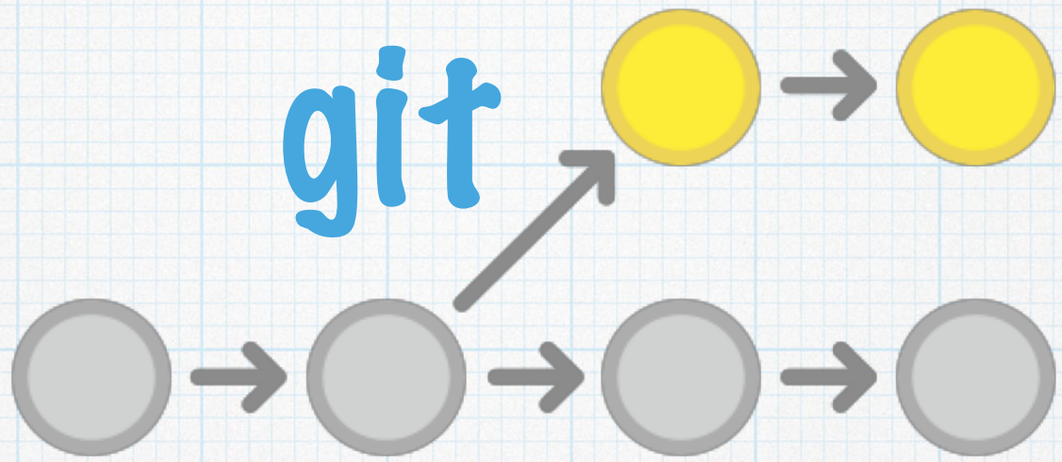


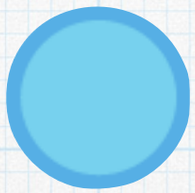
Sistema de Control de Versiones

* Distribuido
* Rápido
* Eficiente (branches)
* Seguro (reflog)
* Flexible

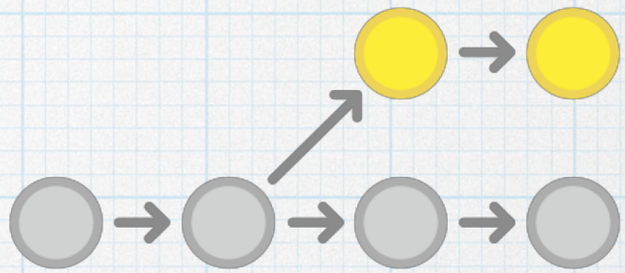
* Local
* Pequeño
* Limpio (~~.svn~~)
* GitHub

git

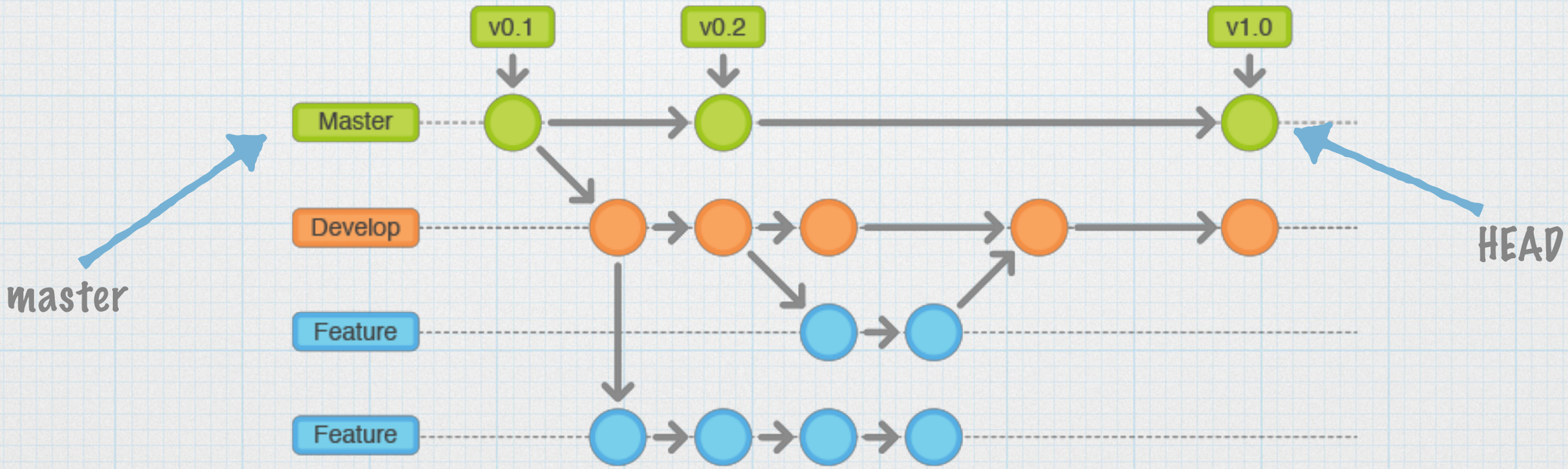
repositorio



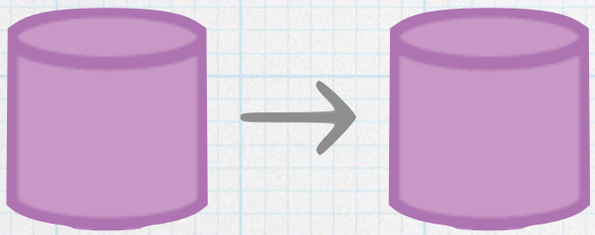
commit



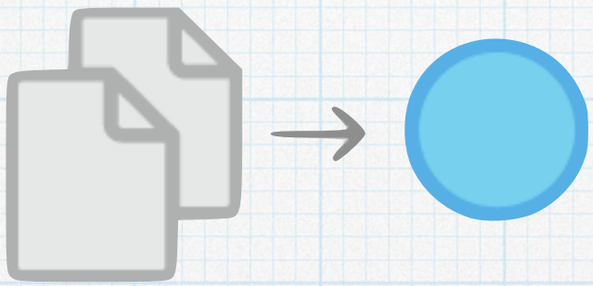
rama



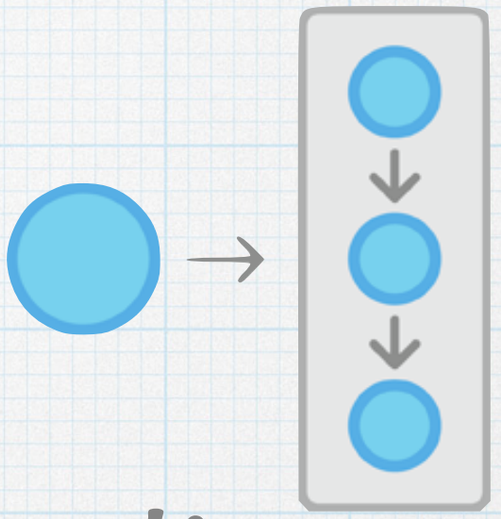
git



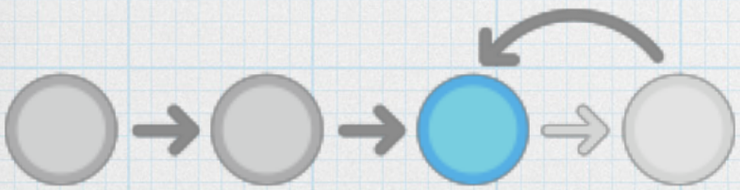
git clone



git add

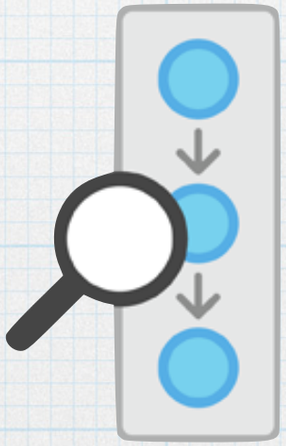


git commit

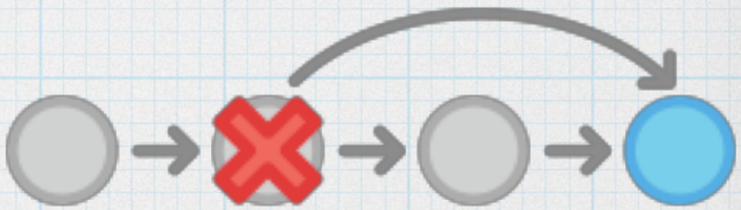


git checkout

git status

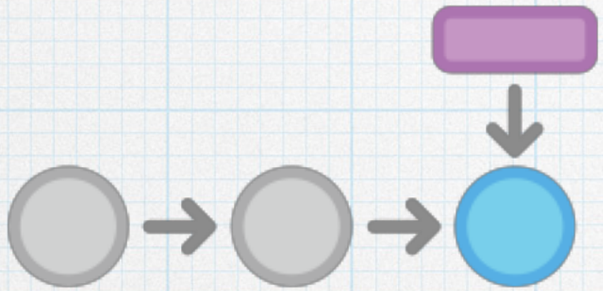


git log

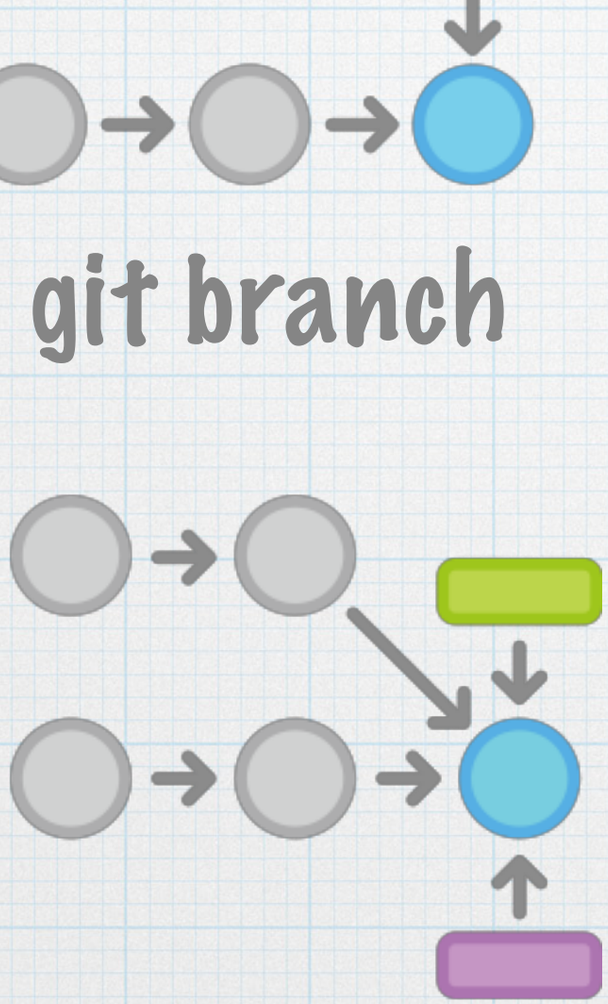


git revert

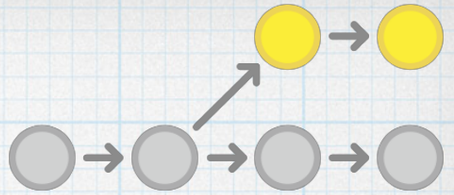
git

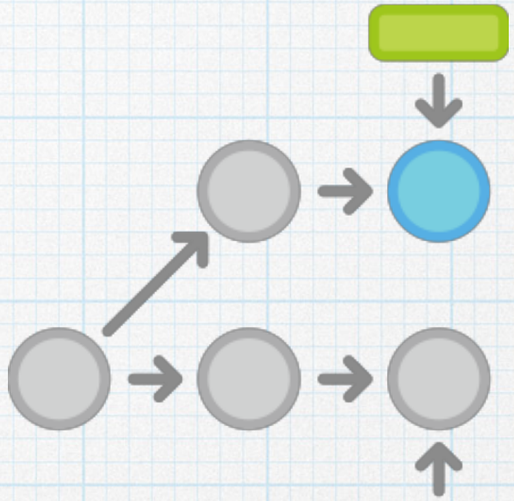


git branch

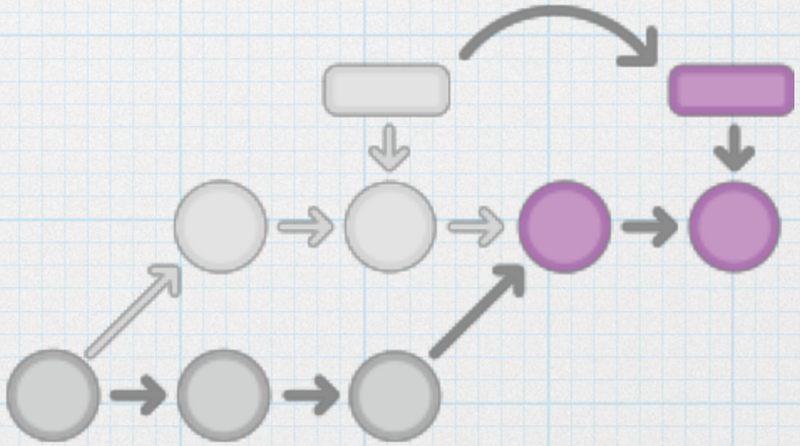


git merge

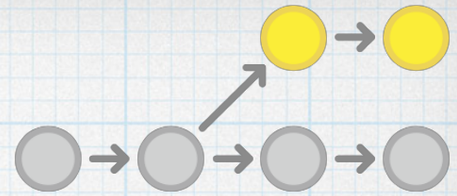




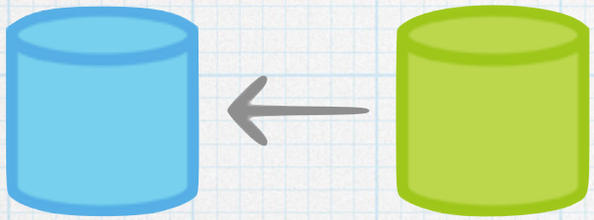
git checkout



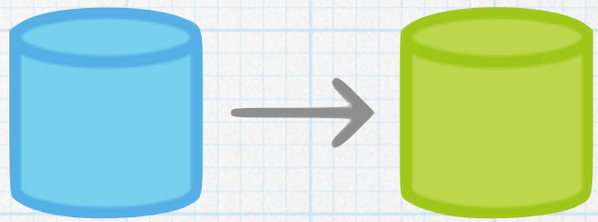
git rebase



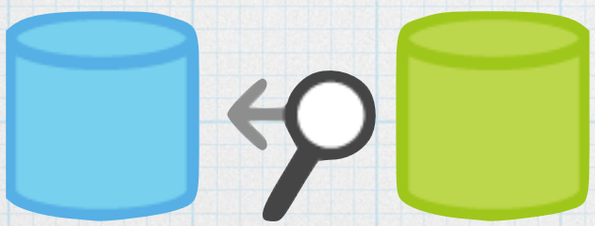
git



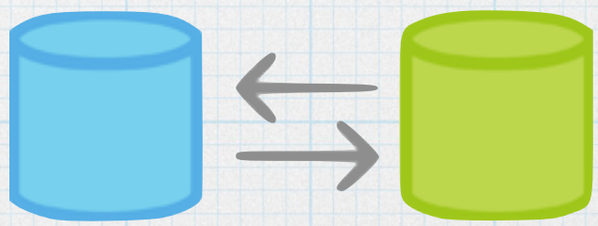
git pull



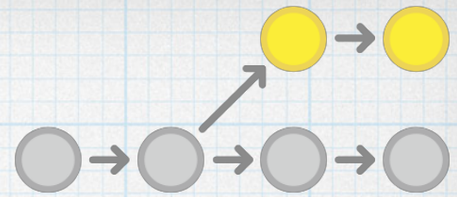
git push



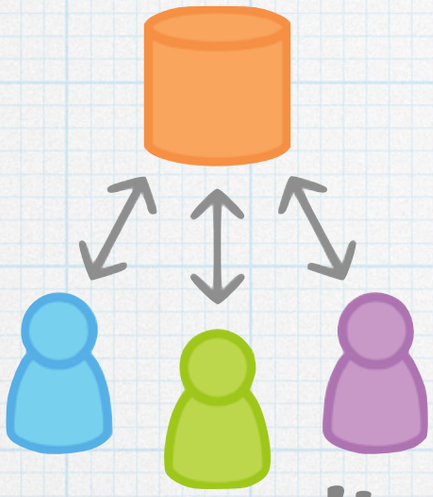
git fetch



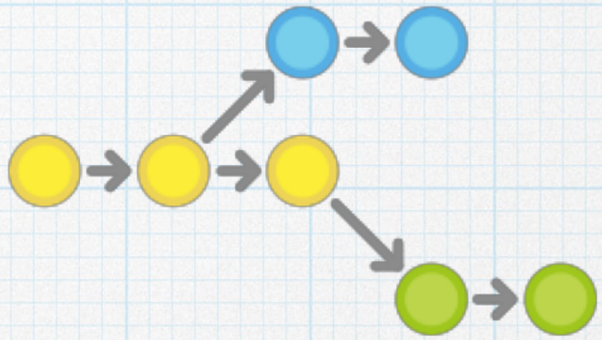
git remote



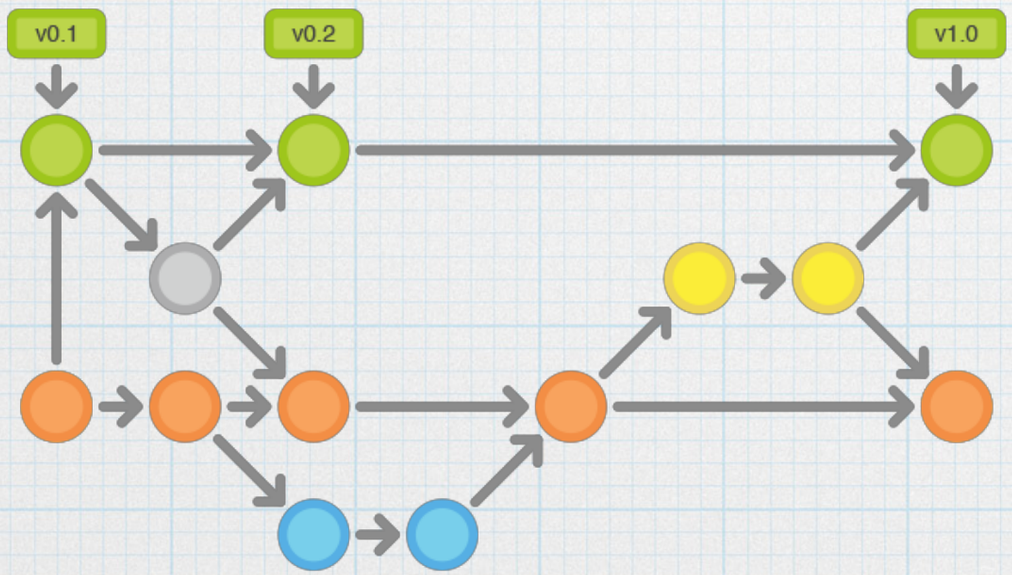
git workflows



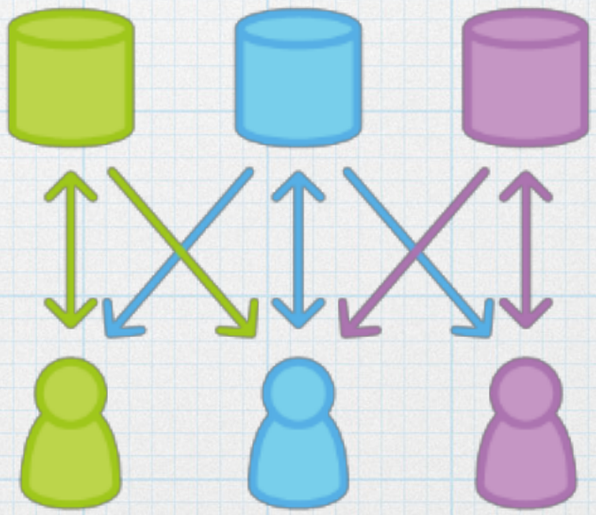
git centralizado



feature branch

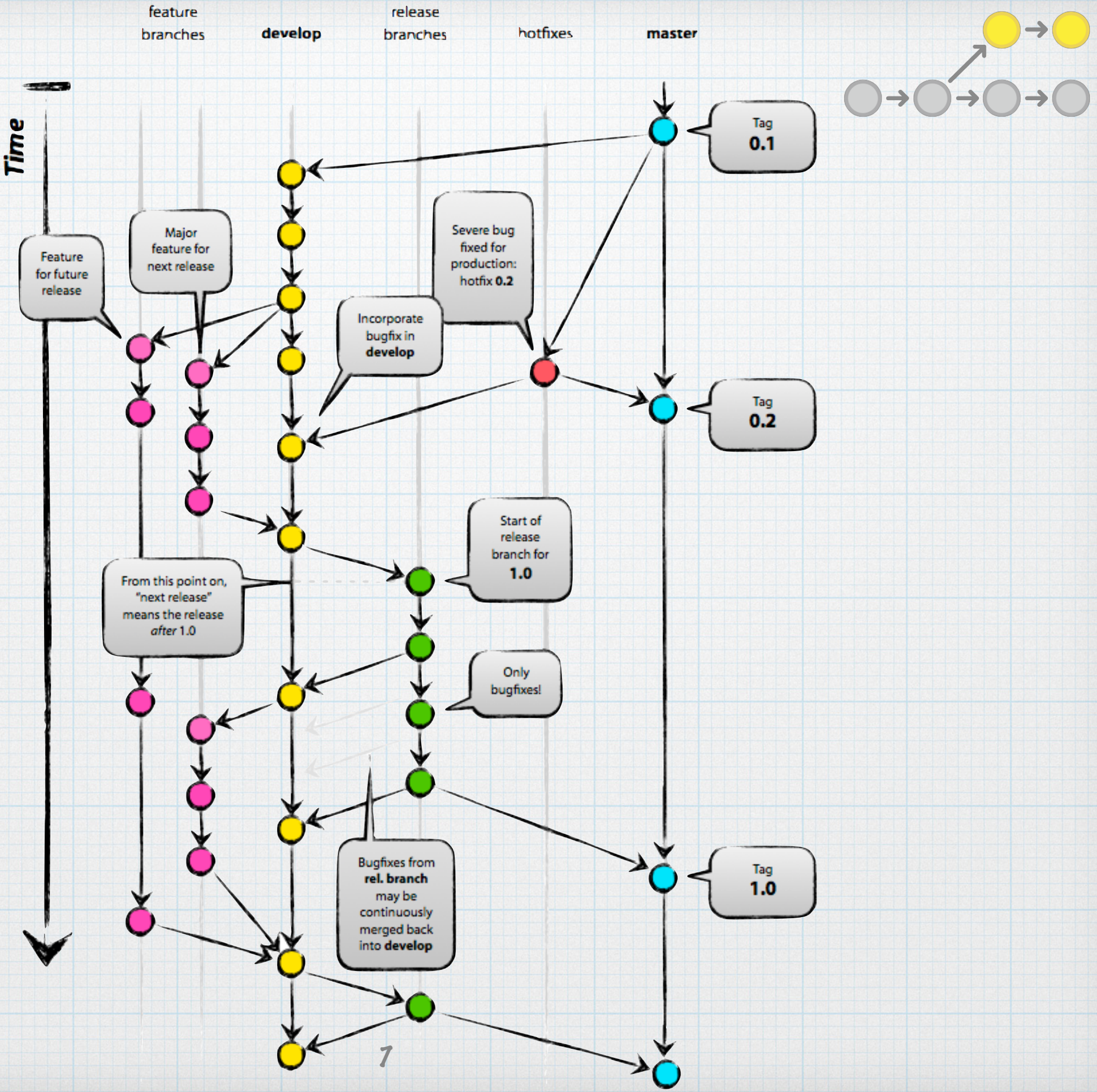


gitflow

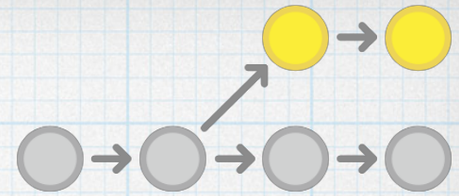


workflow bifurcado

gitflow


feature branches
develop
release branches
hotfixes
master
Time
Feature for future release
Major feature for next release
Incorporate bugfix in develop
Severe bug fixed for production: hotfix 0.2
Tag 0.1
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

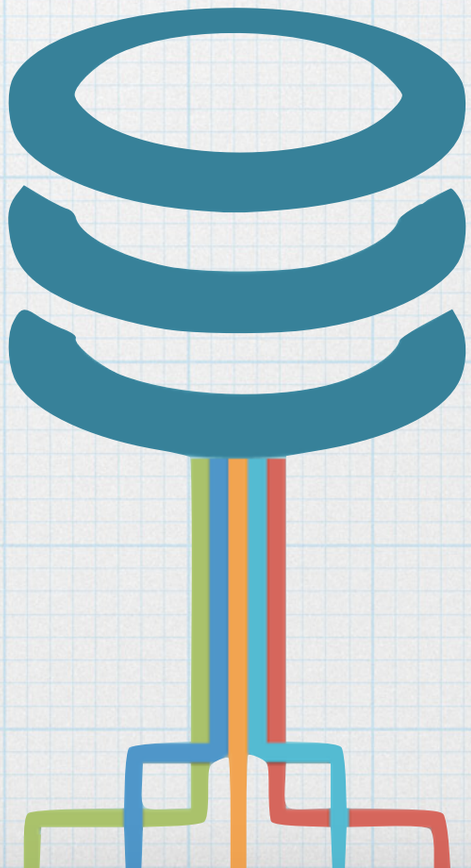
gitflow



```
# OSX
$ brew install git-flow
# Linux
$ apt-get install git-flow
```

```
# start using git flow

$ git flow init
```



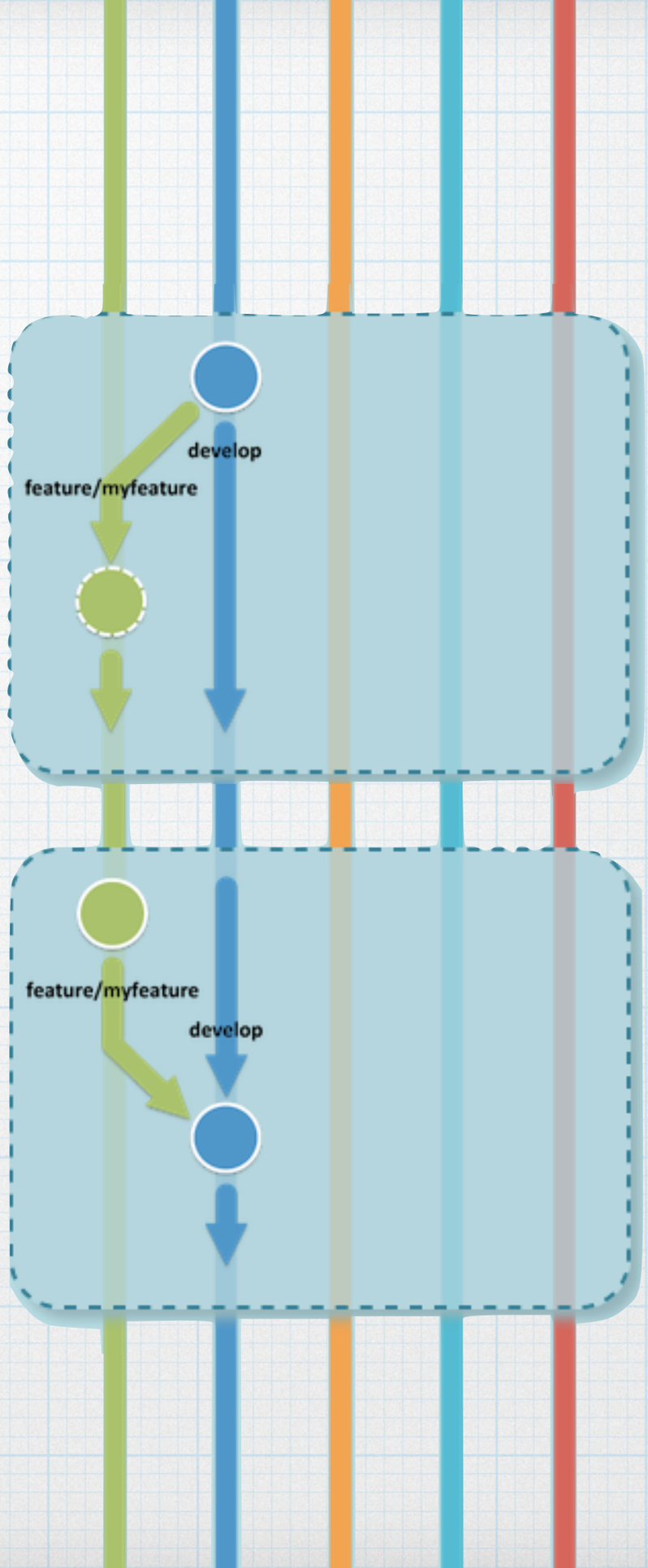
gitflow

```
# start a new feature

$ git flow feature start
MYFEATURE
```

```
# finish up a feature

$ git flow feature finish
MYFEATURE
```



gitflow

```
# publish a feature

$ git flow feature publish
MYFEATURE
```

```
# getting a published feature

$ git flow feature pull
origin MYFEATURE
```

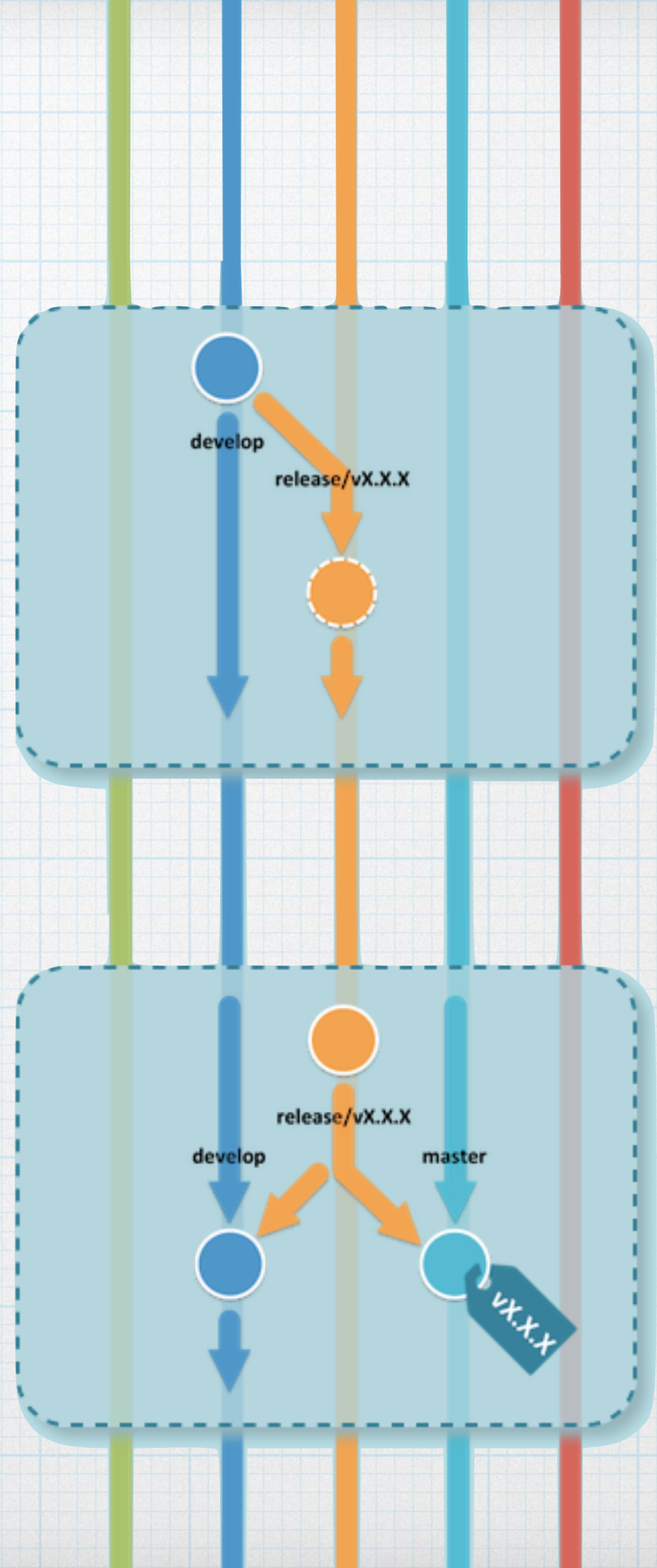
gitflow

```
# start a release
$ git flow release start
MYRELEASE [BASE]
```

```
# publish a release
$ git flow release publish
MYRELEASE
$ git flow release track MYRELEASE
```

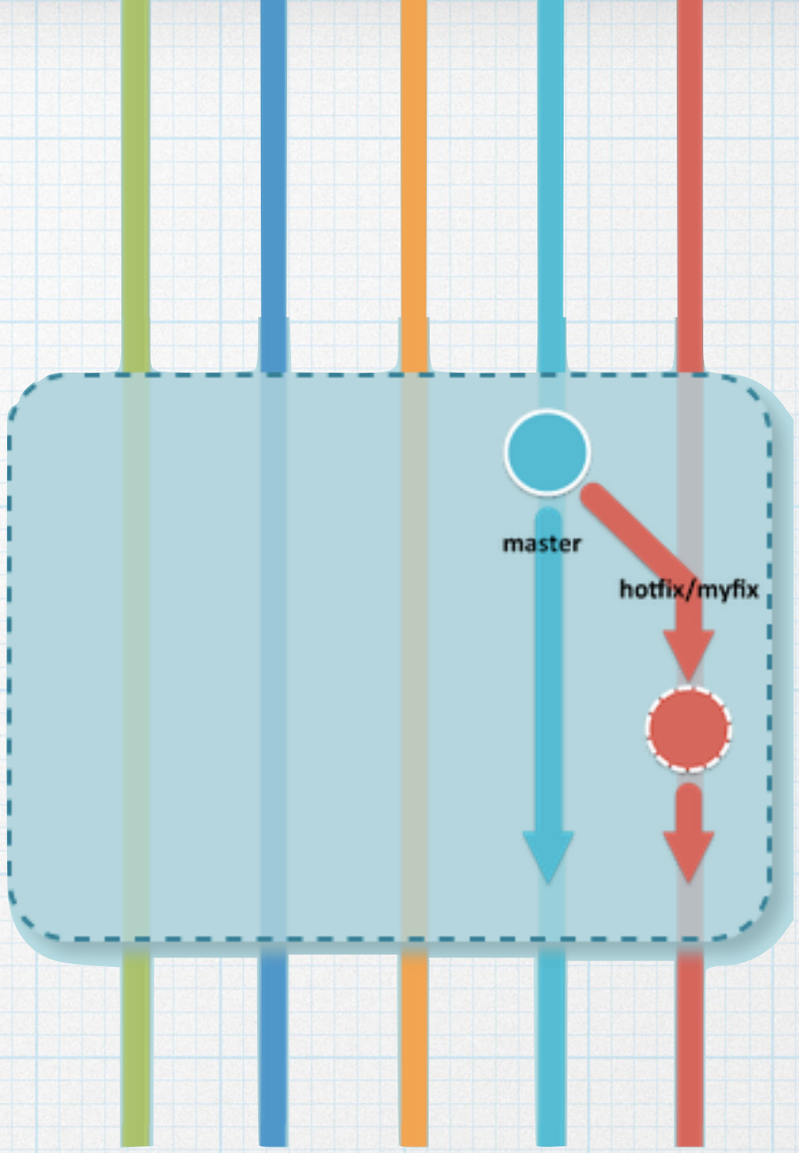
```
# finish up a release

$ git flow release finish
MYFEATURE
$ git push --tags
```



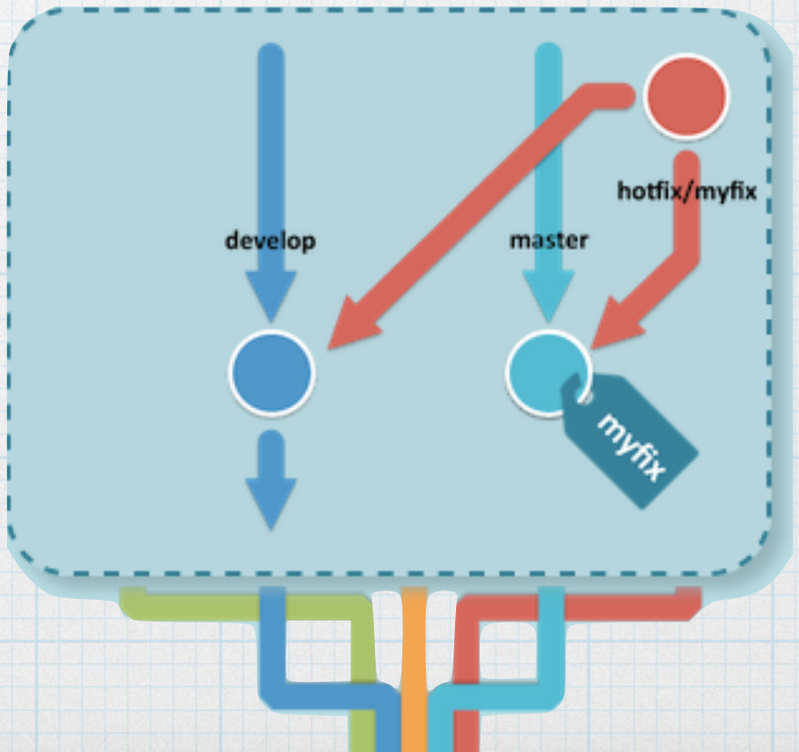
gitflow

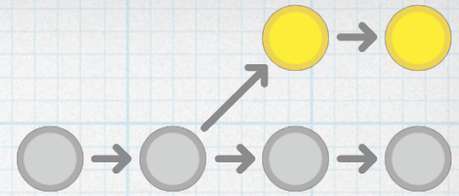
```
# start a hotfix
$ git flow hotfix start
VERSION [BASENAME]
```




```
# finish a hotfix

$ git flow hotfix finish
VERSION
```





git

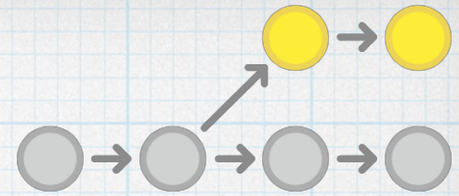
```
# instalar git
$ apt-get install git

# descargar el repositorio
$ git clone https://gitlab.gti-ia.upv.es/...

# realizar un commit
$ git add todos_los_ficheros_modificados
$ git commit -m “mensaje de commit en inglés”

# subir cambios al servidor
$ git push

# bajar cambios del servidor
$ git pull
```



más git

```
# cambiar a una rama
$ git checkout nombre_rama

# cambiar a una rama alojada en el servidor
$ git fetch origin
$ git checkout -b nombre_rama origin/nombre_rama

# ver todas las ramas (la actual marcada con *)
$ git branch

# resolver conflictos
$ git diff                      # complete conflict diff
$ git diff --base $file         # against base file
$ git diff --ours $file         # against your changes
$ git diff --theirs $file       # against other changes
$ git add $conflicting_file     # do for all resolved files
$ git rebase --continue
```