

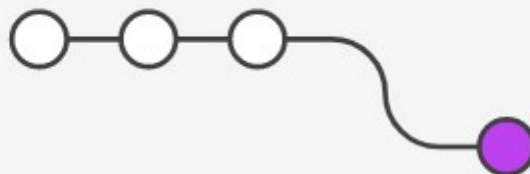


Introducción GIT Flow



• Feature Branch Workflow

Mary begins a new feature



```
git checkout -b marys-feature master
```

```
git status
```

```
git add <some-file>
```

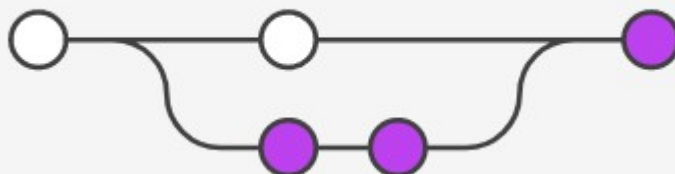
```
git commit -m "Ya agregue feature"
```

```
git push -u origin marys-feature
```



• Feature Branch Workflow

Mary publishes her feature



```
git checkout master
```

```
git merge marys-feature
```



- **GIT FLOW**

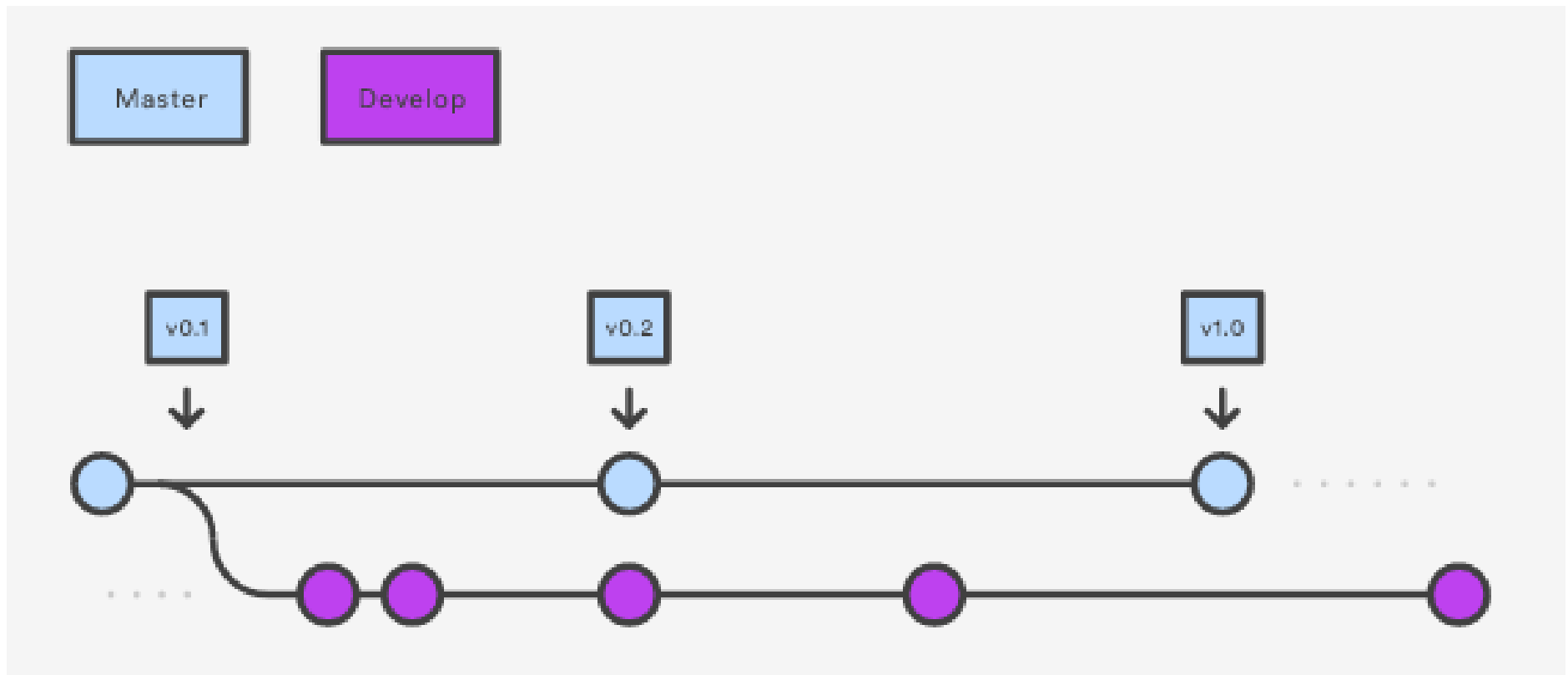
- Asigna roles específicos a cada branch.
- Define cómo interactúan los branches.
- Existe una herramienta que nos gestiona los branches y acciones por medio de comandos.
- Instalación:

```
sudo apt-get install git-flow
```



• GIT FLOW

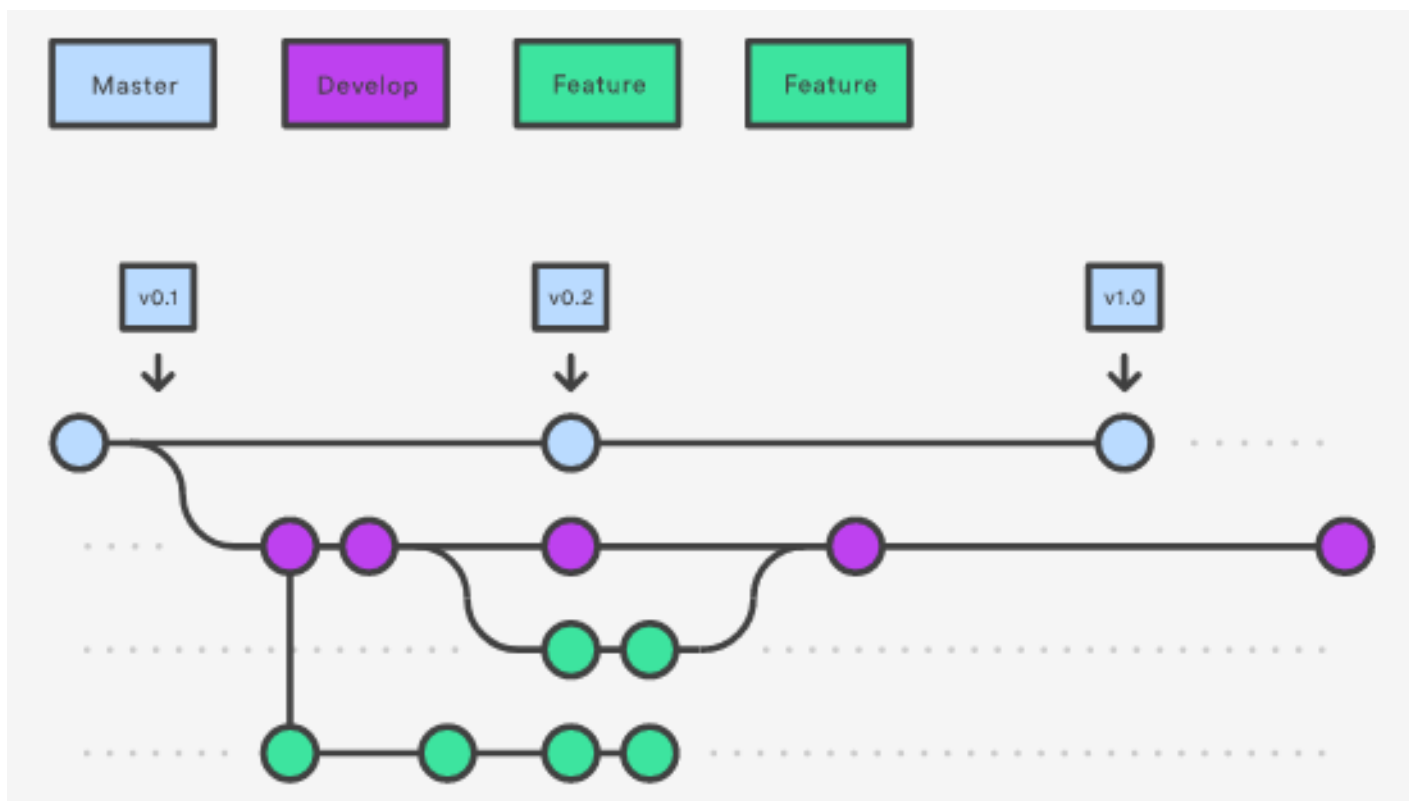
- El branch master se usa para las releases.
- Del branch develop nacen las features.





• GIT FLOW

- Los branches feature nacen de develop
- Los branched feature margean contra develop
- Nunca interactúan con master





• Creación repo

```
git flow init
```

```
Initialized empty Git repository in /home/...  
No branches exist yet. Base branches must be created now.  
Branch name for production releases: [master]  
Branch name for "next release" development: [develop]
```

```
How to name your supporting branch prefixes?
```

```
Feature branches? [feature/]
```

```
Bugfix branches? [bugfix/]
```

```
Release branches? [release/]
```

```
Hotfix branches? [hotfix/]
```

```
Support branches? [support/]
```

```
Version tag prefix? []
```

```
Hooks and filters directory? [/home/.../.git/hooks]
```



• Creación feature

git branch

*develop

master

git flow feature start mary-feature

git branch

develop

*feature/mary-feature

master

• Es equivalente a:

git checkout develop

git checkout -b feature_branch



• Finalización feature

git flow feature finish mary-feature

Summary of actions:

- The feature branch 'feature/mary-feature' was merged into 'develop'
- Feature branch 'feature/mary-feature' has been locally deleted
- You are now on branch 'develop'

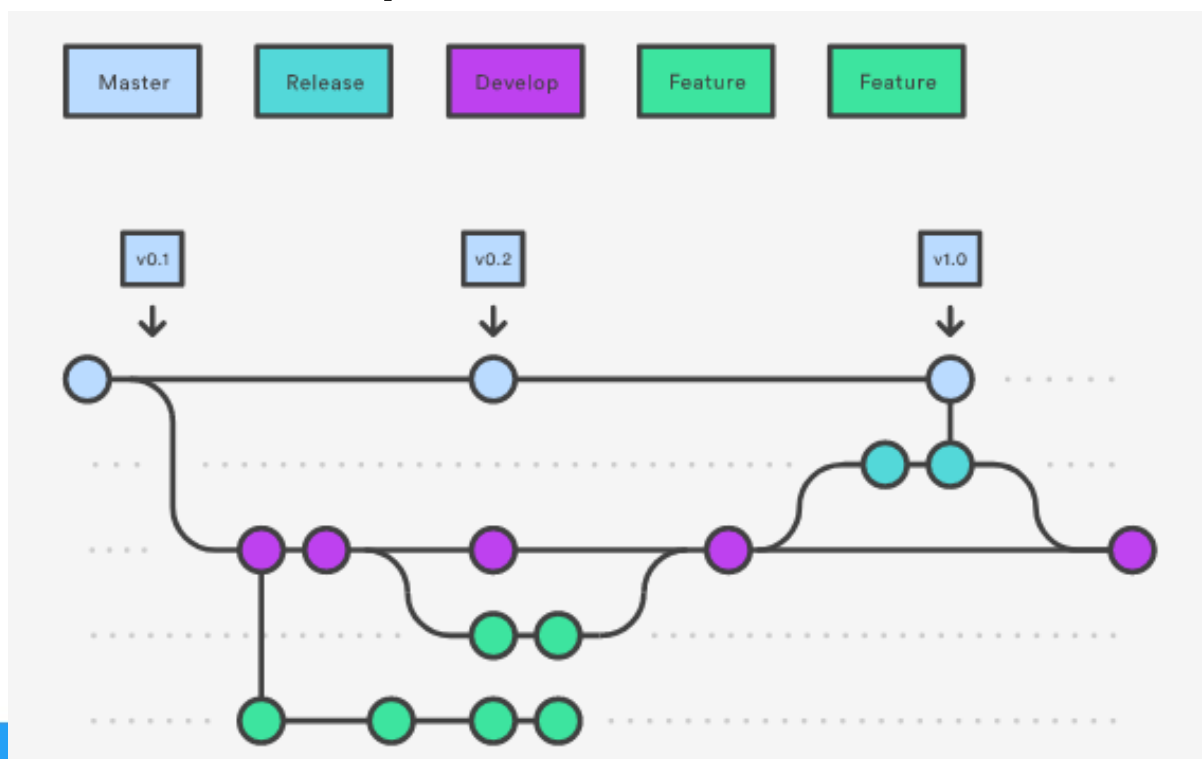
• Es equivalente a:

```
git checkout develop  
git merge mary-feature
```



• Preparación release

- Cuando develop tiene todas las features para una release, creamos un branch release.
- Este branch release es el que mergea con master
- Permite hacer ajustes finales a la release mientras se sigue trabajando en develop.





- **Inicio de release**

```
git flow release start 0.1.0
```

```
Switched to a new branch 'release/0.1.0'
```

- **Es equivalente a:**

```
git checkout develop
```

```
git checkout -b release/0.1.0
```



• Finalización de release

```
git flow release finish '0.1.0'
```

- Se hace un merge contra master y develop, ya que en el branch release se pudo haber hecho un cambio que no estaría en develop.

• Es equivalente a:

```
git checkout master  
git merge release/0.1.0
```

```
git checkout develop  
git merge release/0.1.0
```



• RESUMEN

```
git flow init
```

```
git add -A  
git commit -a -m "Primer commit"
```

```
git flow feature start mi_feature
```

```
git add -A  
git commit -a -m "Commit de feature"
```

```
git flow feature finish mi_feature
```

...

```
git flow release start 0.1.0  
git flow release finish '0.1.0'
```



Bibliografía

- <https://www.atlassian.com/en/git/tutorials/comparing-workflows/gitflow-workflow>
- <https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow>