

git-flow





Ingredients:
Pork with Ham,
Salt, Water,
Modified Potato
Starch, Sugar,
Sodium Nitrite

SPAM[®]

Classic

U.S.
INSPECTED
AND PASSED BY
DEPARTMENT OF
AGRICULTURE

Serving
Suggestion

NET WT
12 OZ
(340g)



jesus@jesusamieiro.com





QUADRALIA



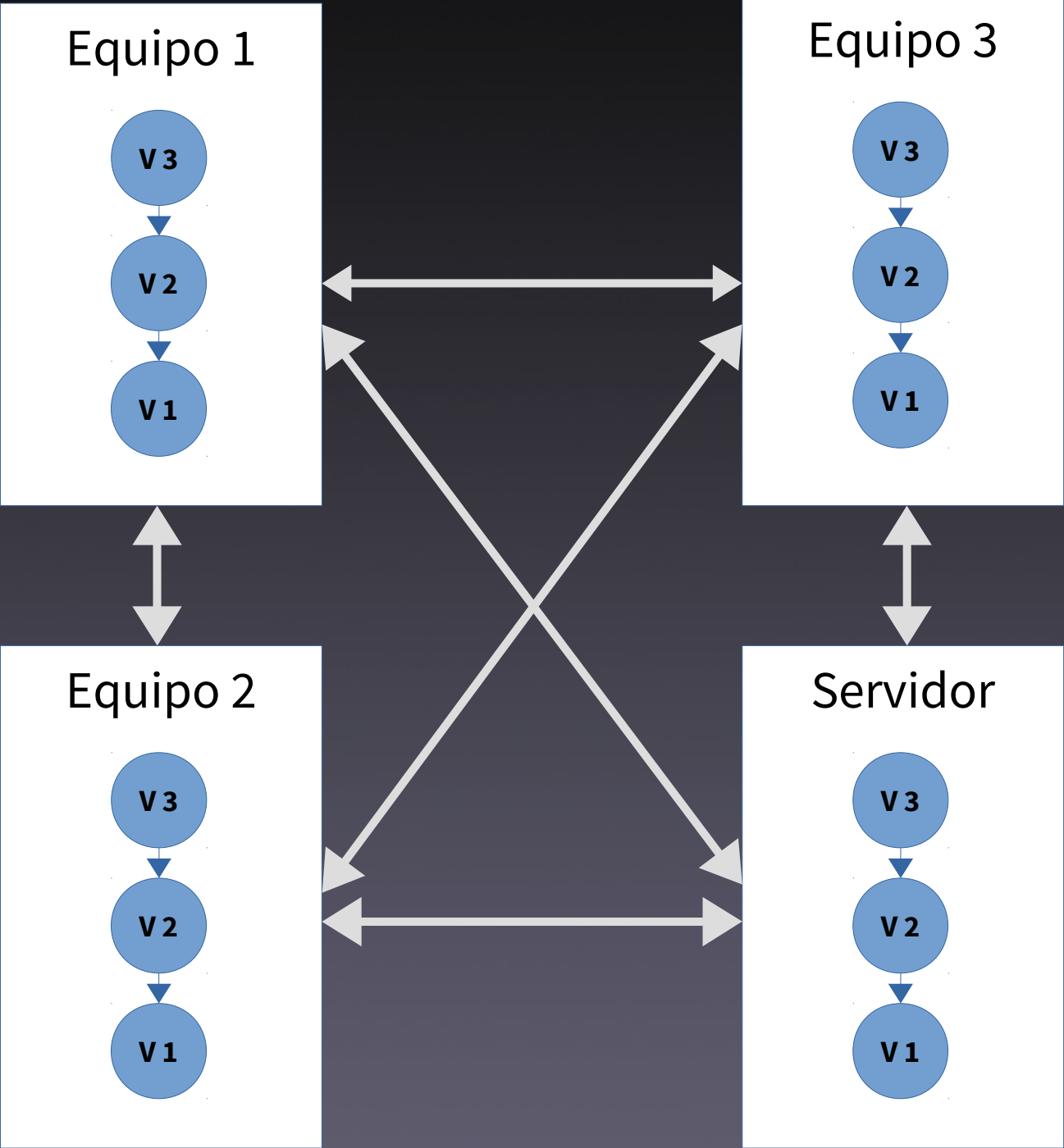
Contido

- Refresco conceptos Git
- Git-flow
 - Introducción
 - Fluxos de traballo
 - Instalación
 - Comandos de git-flow
- Exercicio

git

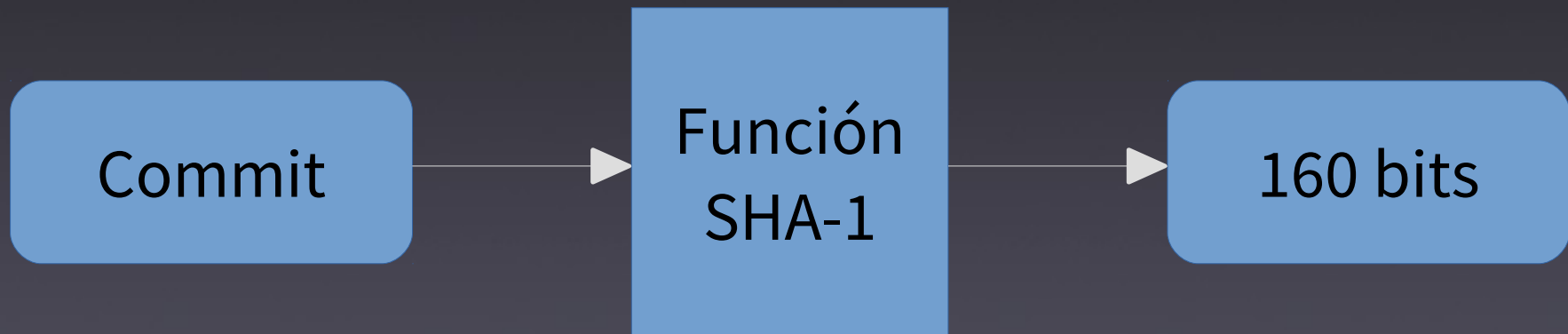
Distribuído





Commit

SHA-1



HEAD

Commit 1



Commit 1



Commit 2



Commit 1



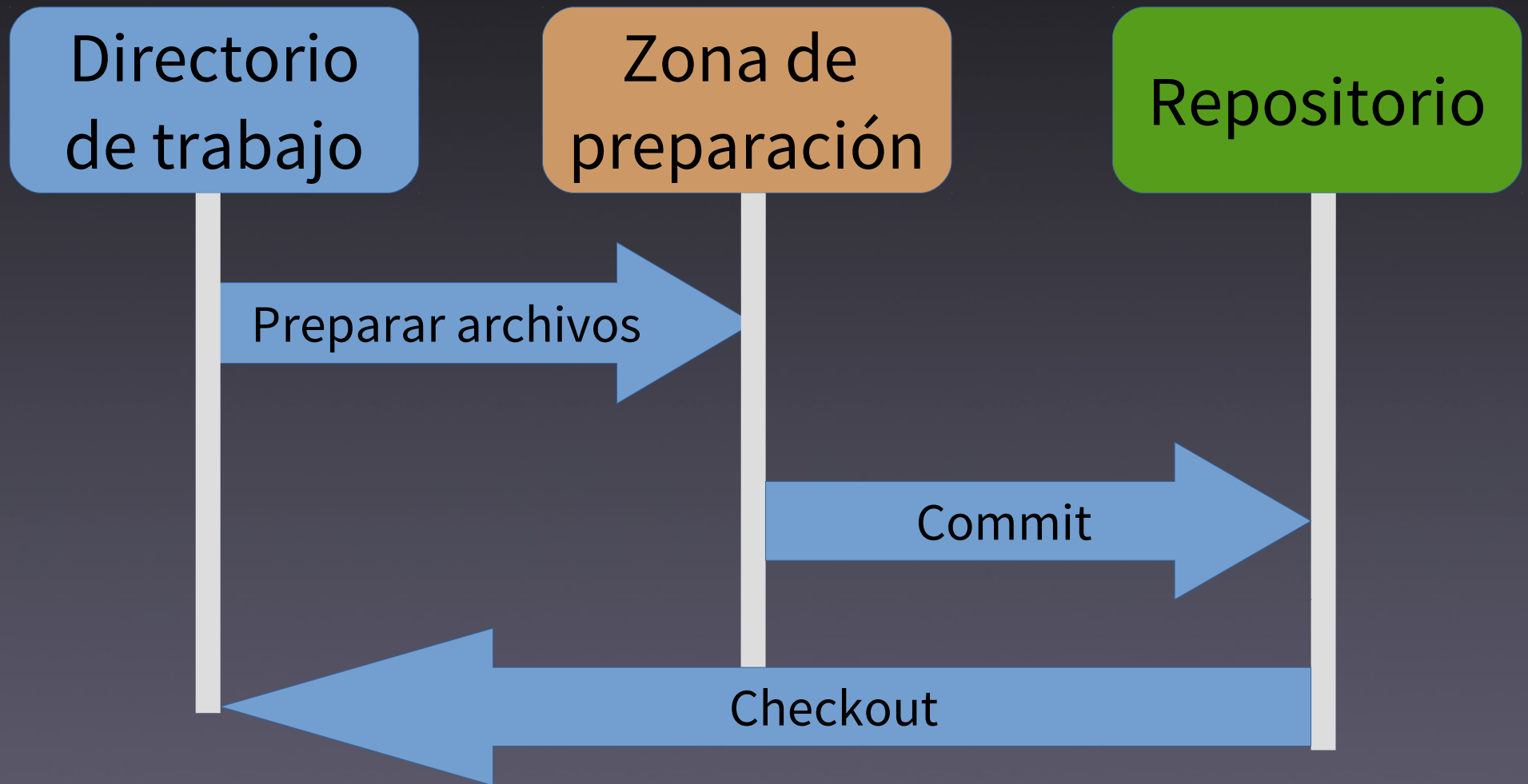
Commit 2



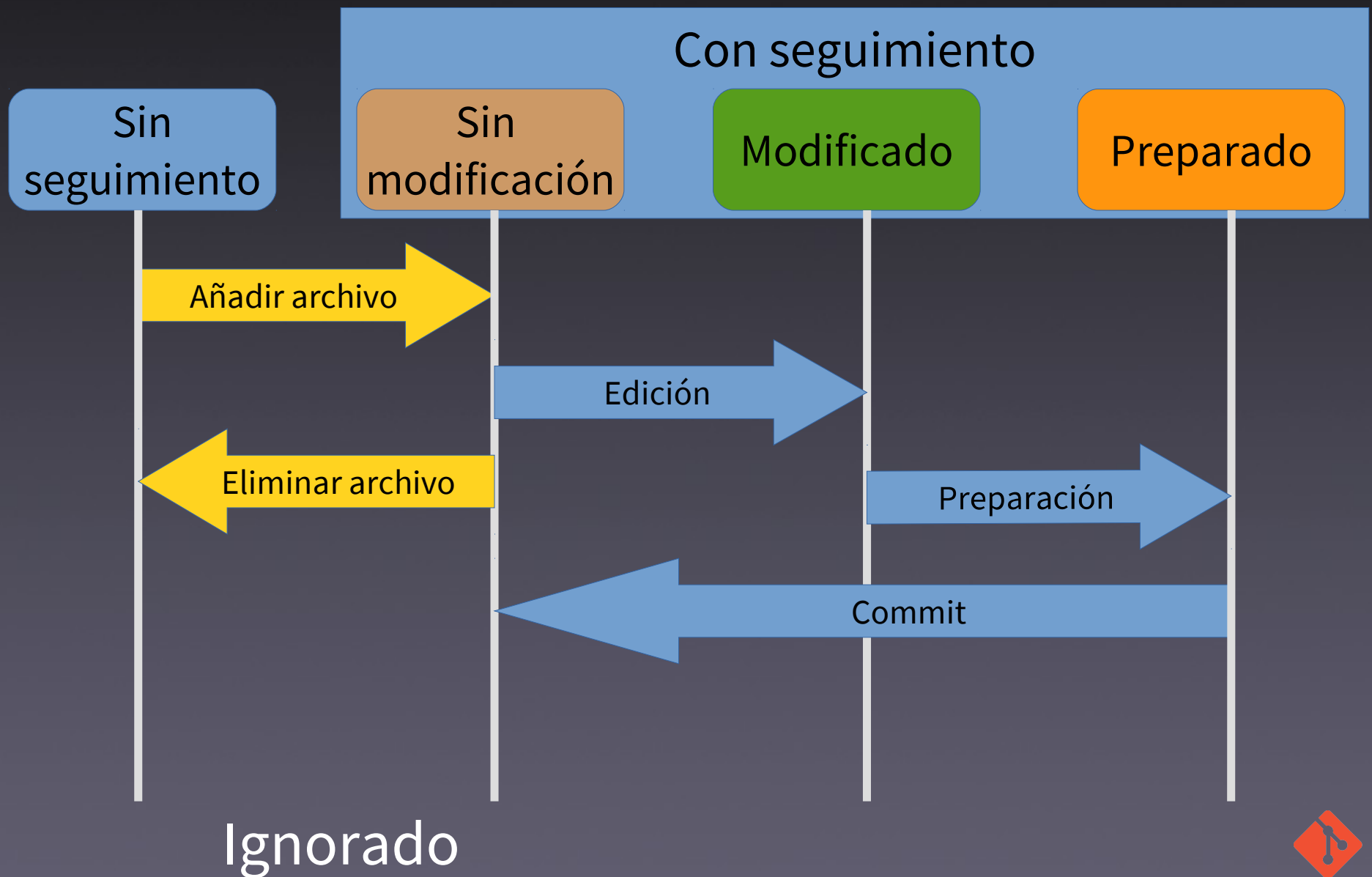
Commit 3



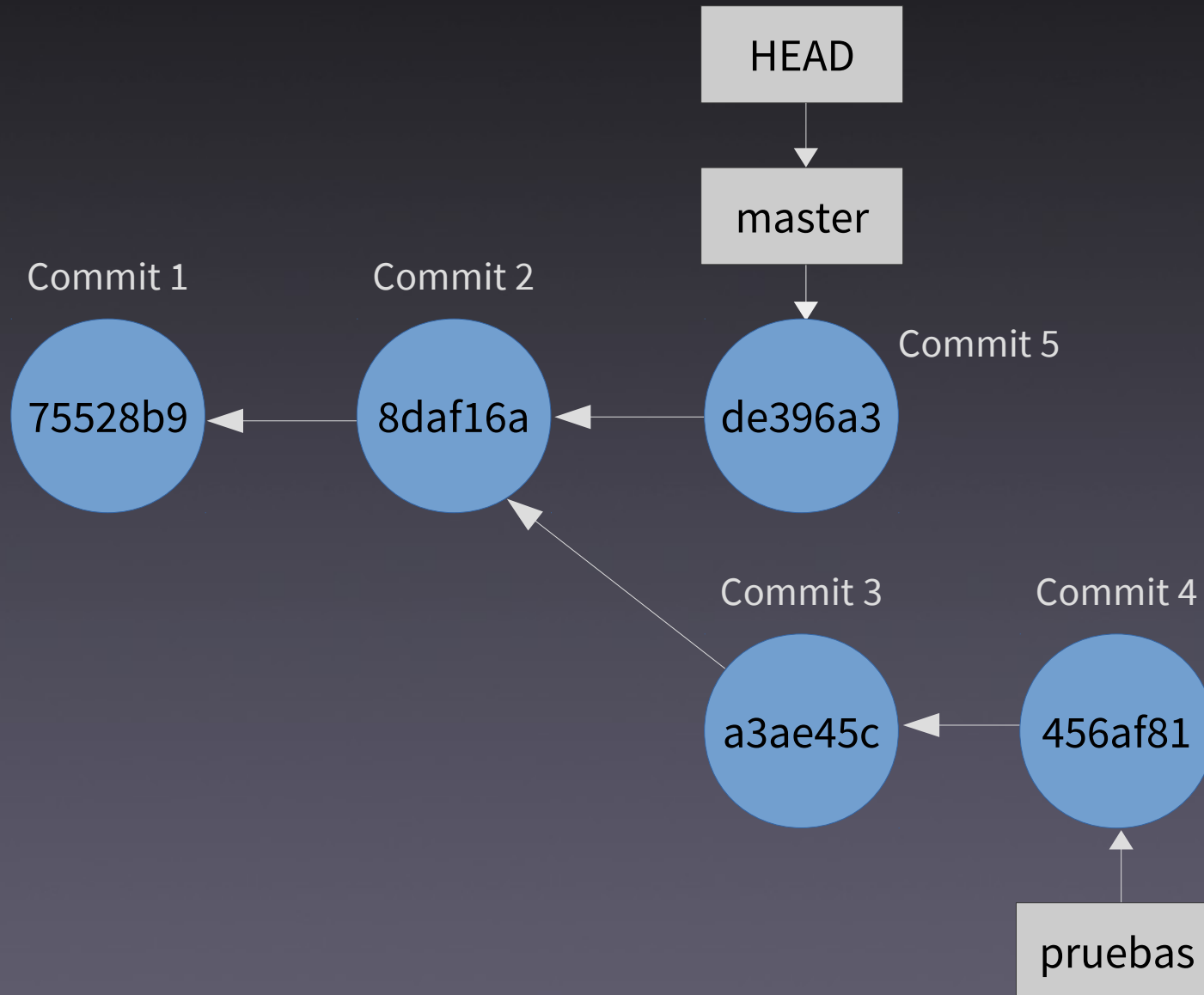
Zonas en Git



Estados e flujo



Ramas



Repositorios remotos

Repositorios remotos

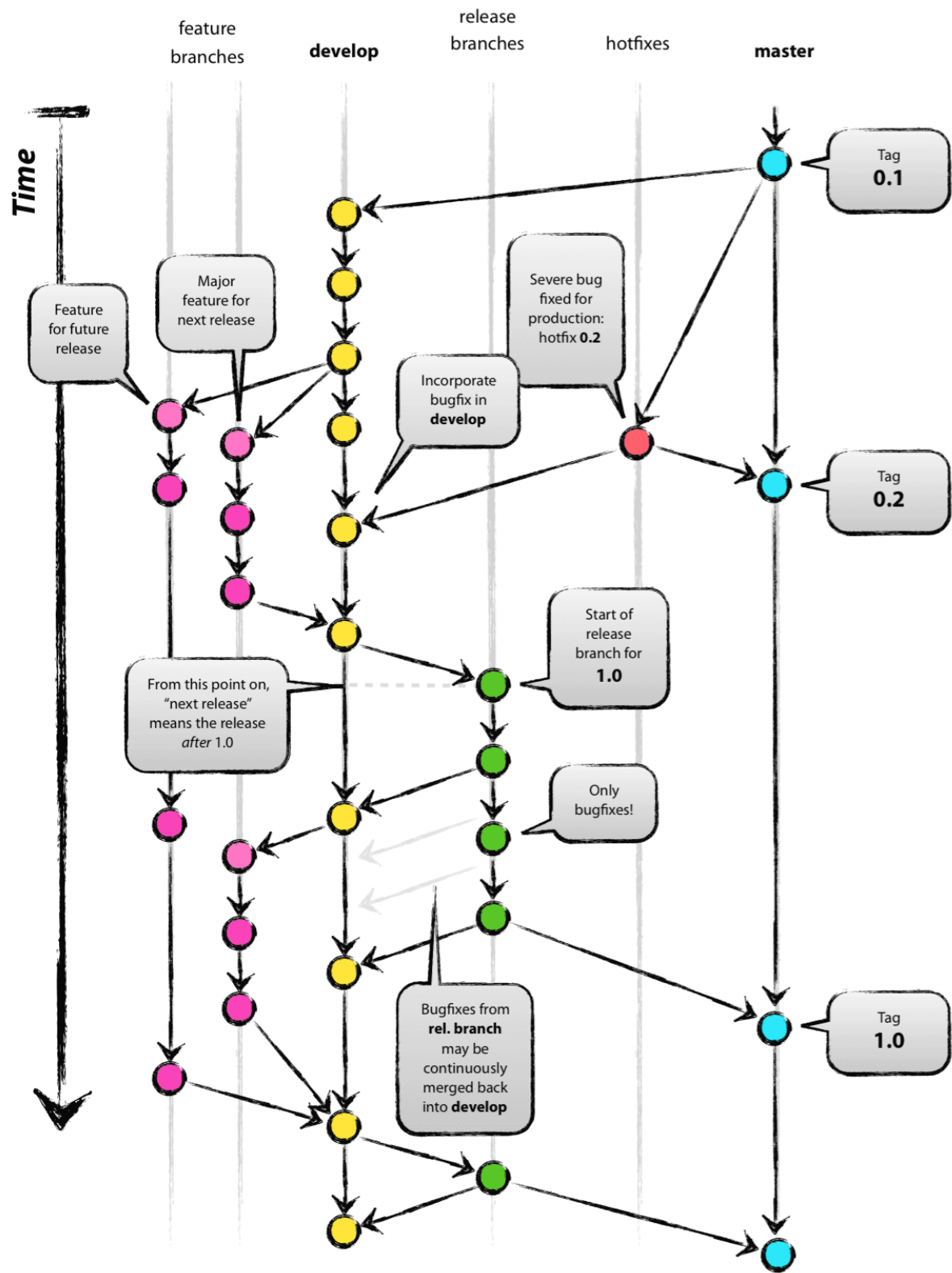
git remote

git push

git fetch

git pull

git-flow



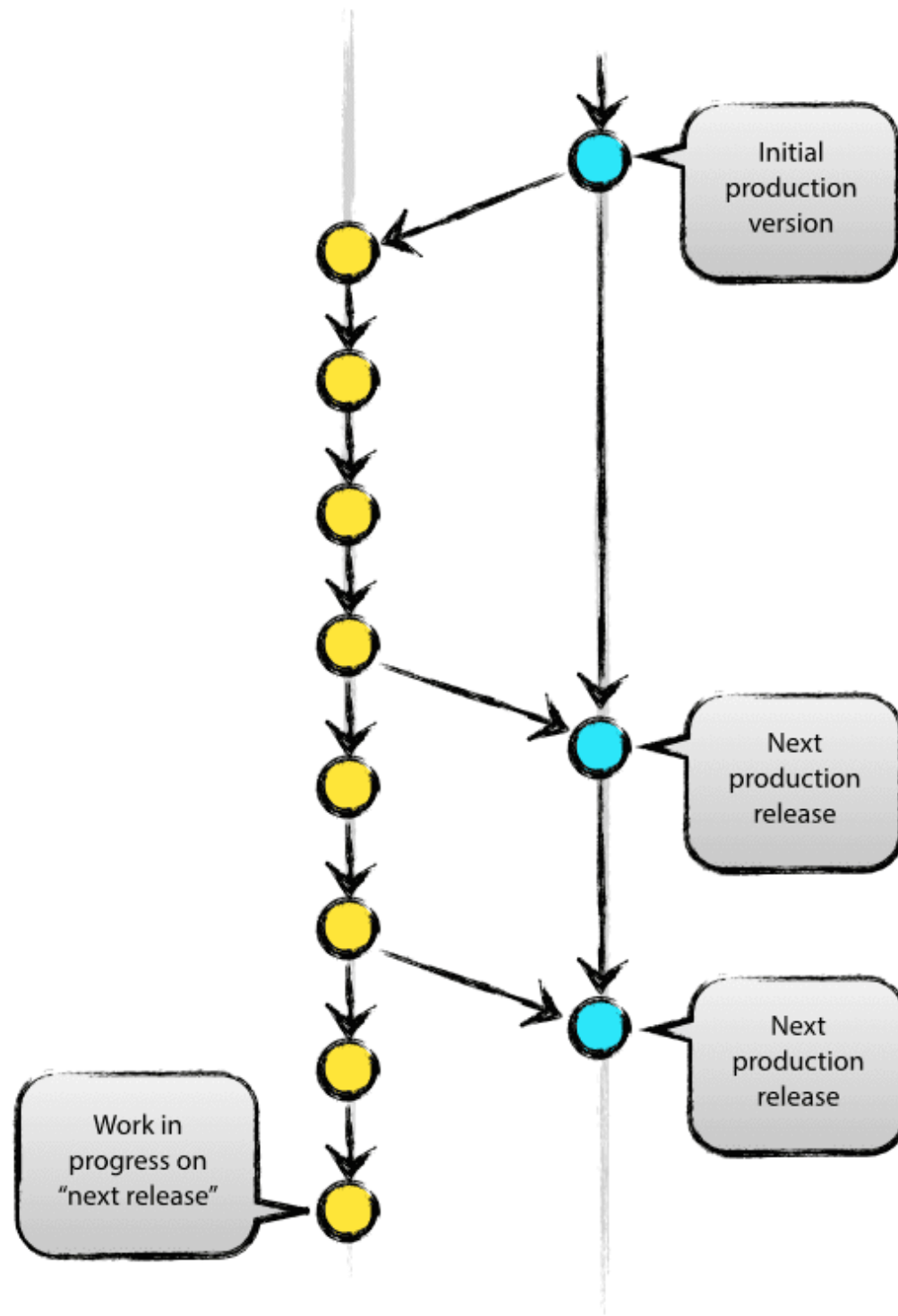
Ramas principais

- master
- develop

- Permanentes

develop

master



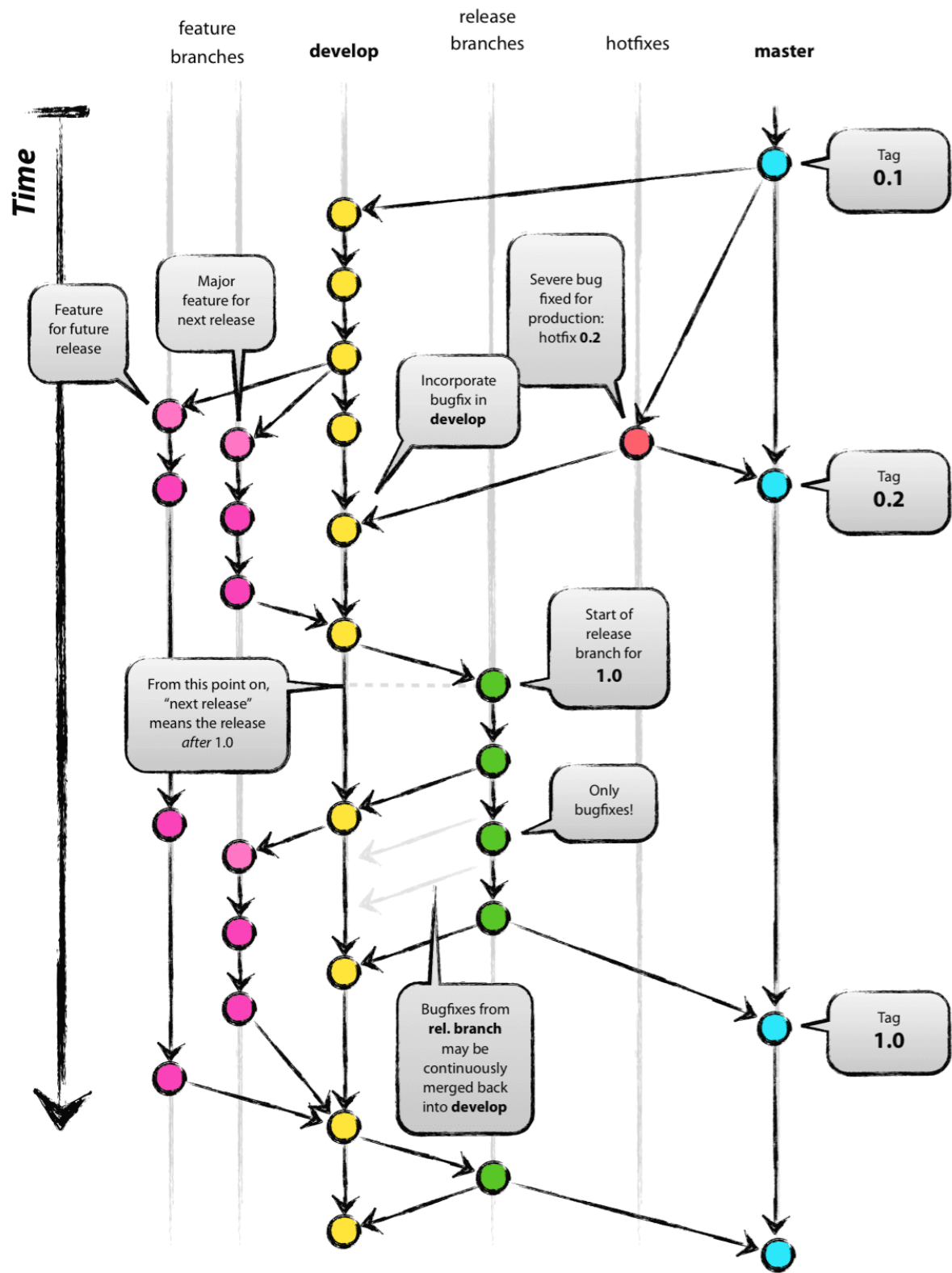
Ramas de soporte

- Feature
- Release
- Hotfix

- Bugfix
- Support

Ramas de suporte

- Propósito determinado
- Conjunto de regras de uso
- Tempo de vida limitado
- Ramas normais

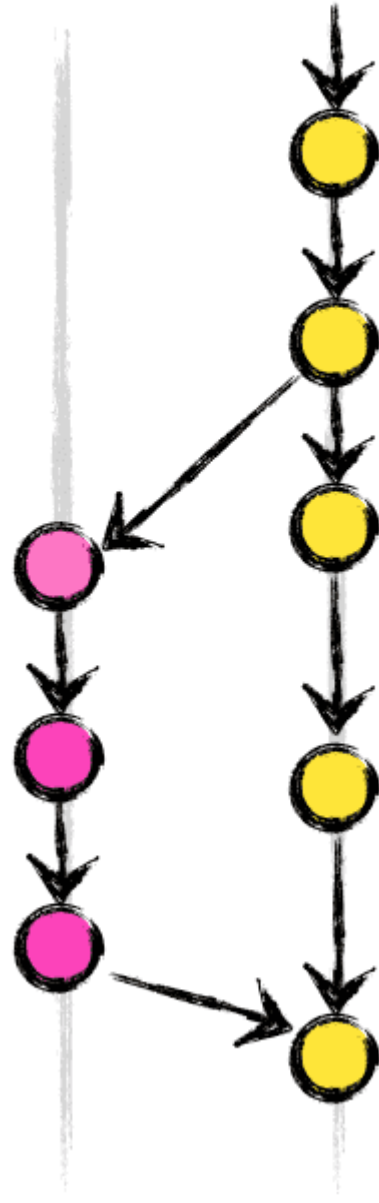


Ramas feature (topic)

- Créanse a partir de develop
- Fusiónanse en develop
- Nome: calquera, excepto:
 - master
 - develop
 - release-*
 - hotfix-*
- Obxectivo: novas funcionalidades

feature
branches

develop



Ramas feature (topic)

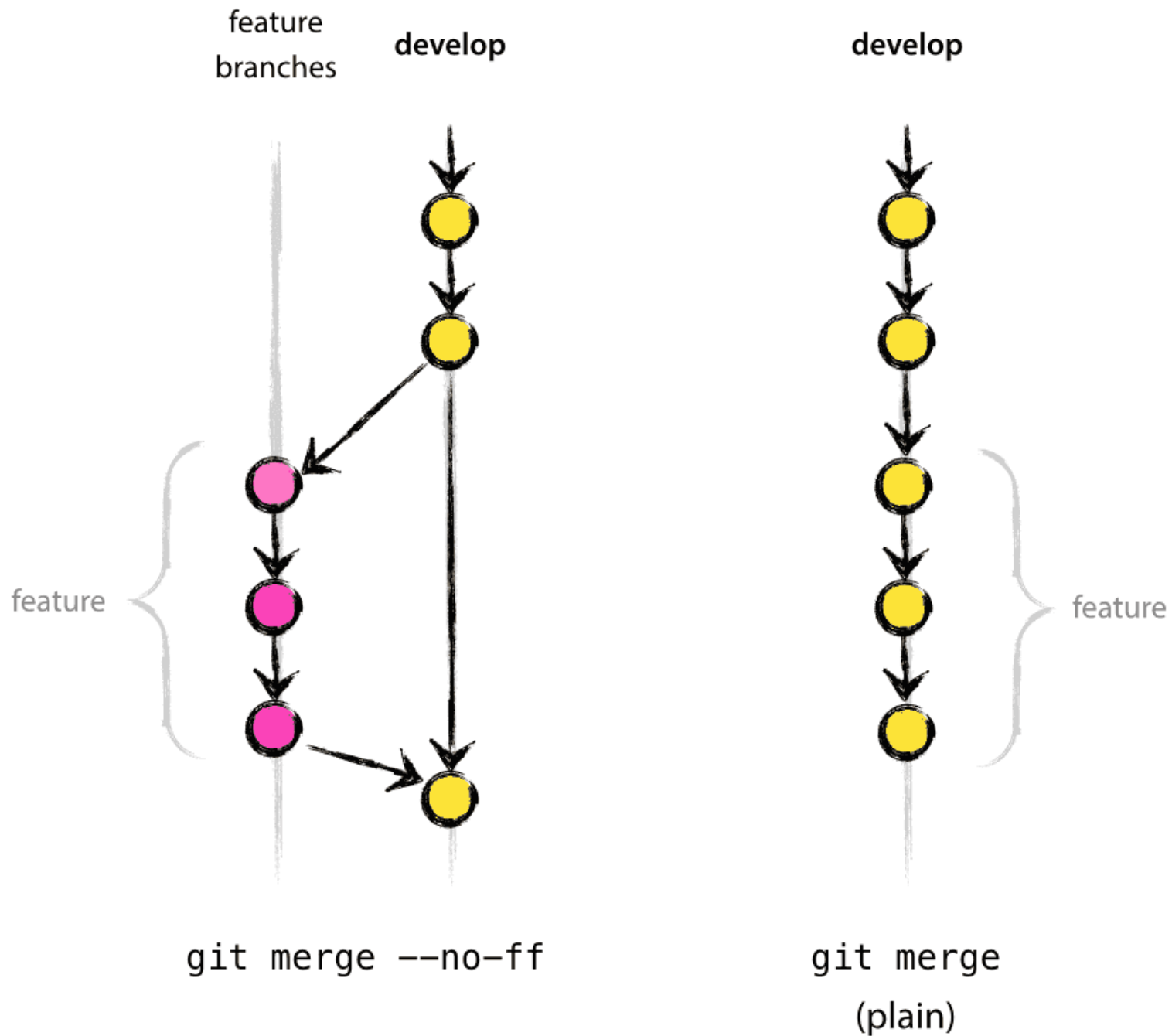
```
git checkout -b nova_funcionalidade develop
```

```
git checkout develop
```

```
git merge --no-ff nova_funcionalidade
```

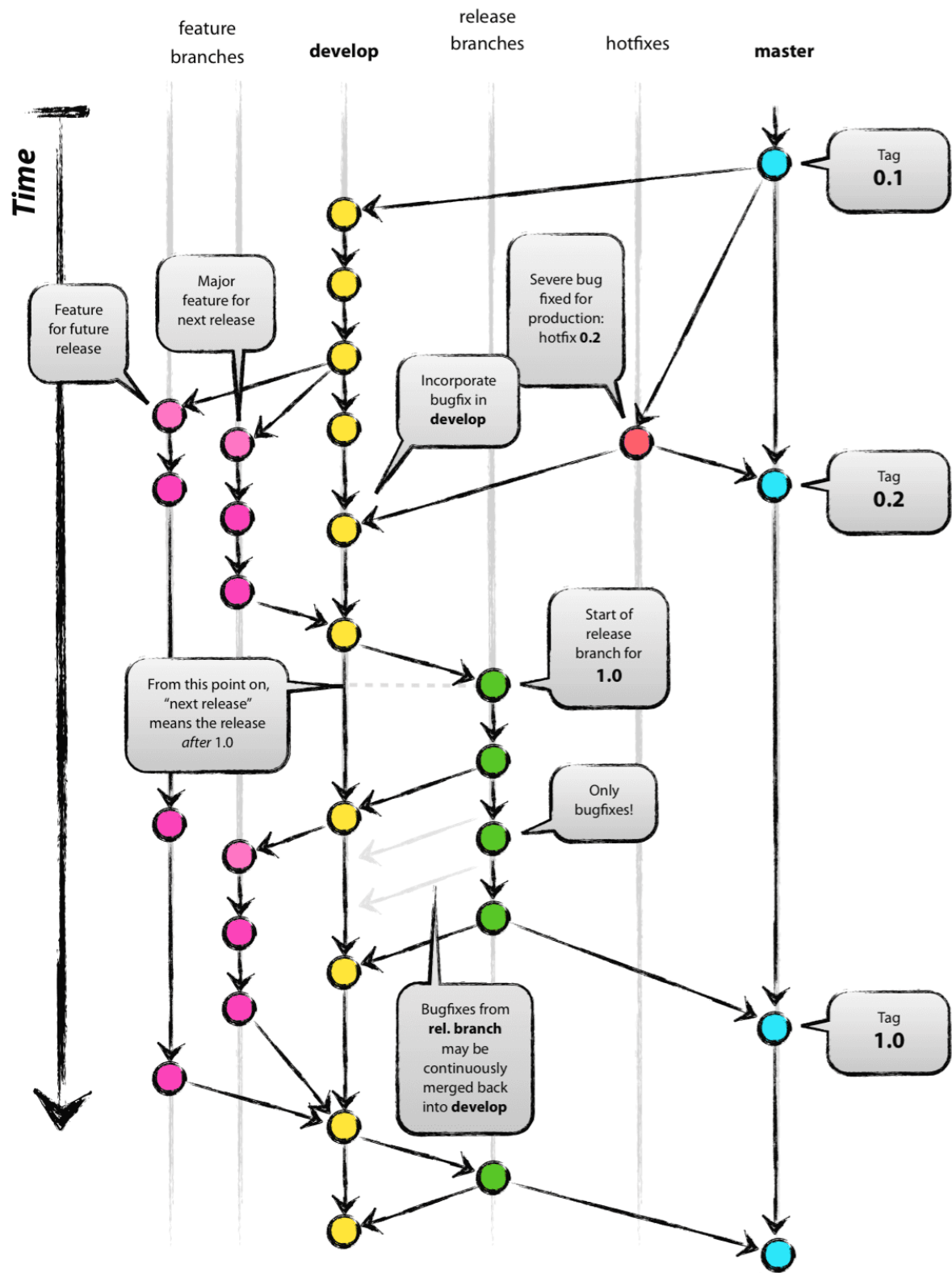
```
git branch -d nova_funcionalidade
```

```
git push origin develop
```



Ramas release

- Créanse a partir de develop
- Fusiónanse en master e en develop
- Nome: release-*
- Obxectivo: preparación dunha nova publicación (release)



Ramas release

```
git checkout -b release-2.6 develop
```

```
git checkout master
```

```
git merge --no-ff release-2.6
```

```
git tag -a 2.6
```

```
git checkout develop
```

```
git merge --no-ff release-2.6
```

```
git branch -d release-2.6
```

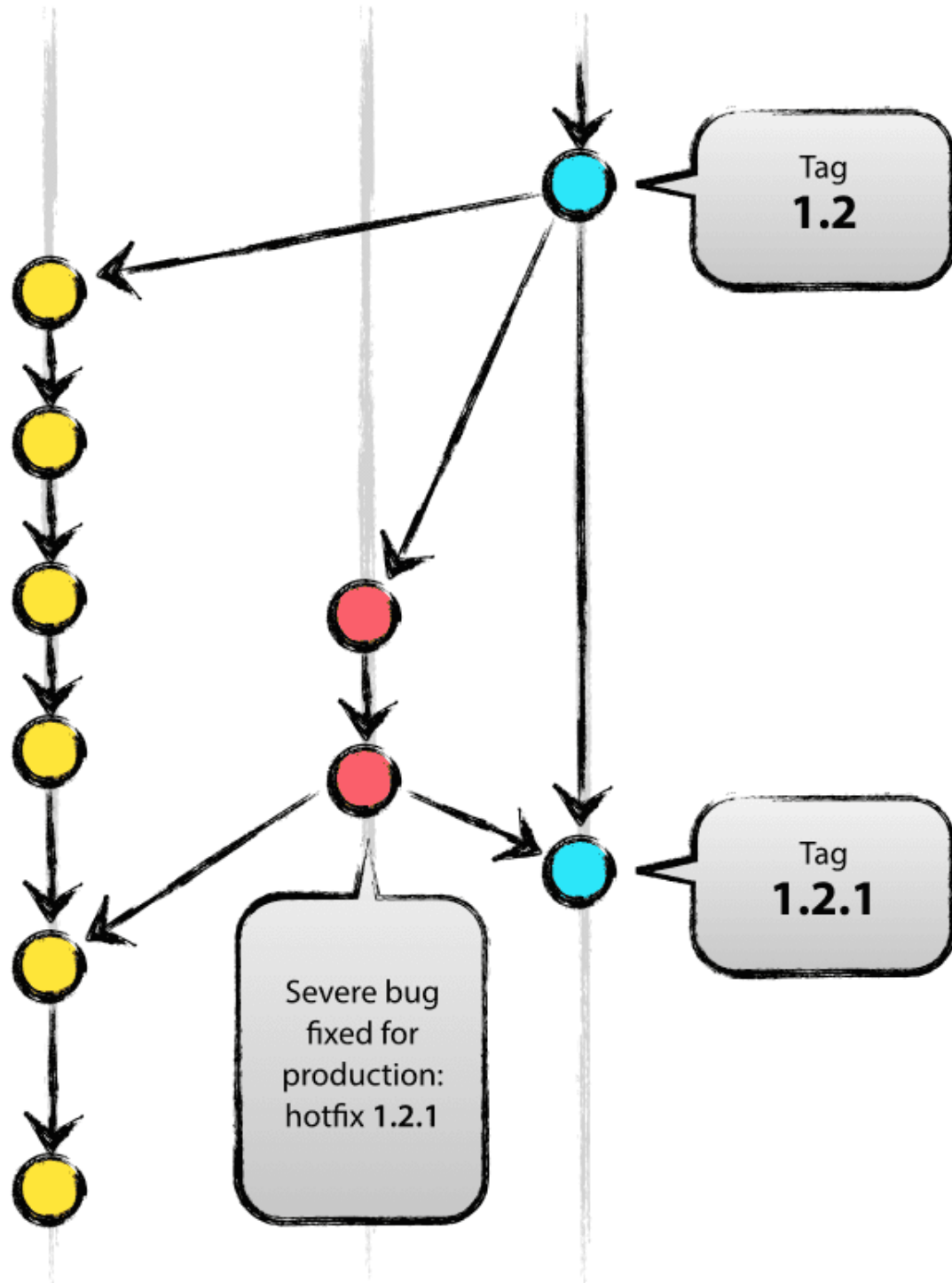
Ramas hotfix

- Créanse a partir de master
- Fusiónanse en master e en develop
- Nome: hotfix-*
- Obxectivo: resolución de erros en produción

develop

hotfixes

master



Ramas hotfix

```
git checkout -b hotfix-2.6.1 master
```

```
git checkout master
```

```
git merge --no-ff hotfix-2.6.1
```

```
git tag -a 2.6.1
```

```
git checkout develop
```

```
git merge --no-ff hotfix-2.6.1
```

```
git branch -d hotfix-2.6.1
```

Ramas bugfix

- Créanse a partir de develop
- Fusiónanse en develop
- Nome: bugfix-*
- Obxectivo: resolución de erros en desenvolvemento

Ramas bugfix

```
git checkout -b bugfix-login develop
```

```
git checkout develop
```

```
git merge --no-ff bugfix-login
```

```
git branch -d bugfix-login
```

Ramas support

- Créanse a partir de master
- Non se integran noutra rama nin se eliminan
- Nome: support-*
- Obxectivo: soporte a sistemas en produción con versións anticuadas

Ramas support

```
git checkout -b support-v2.0-udc 2.0.0
```



Instalación

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

```
sudo dnf install gitflow
```

```
brew install git-flow-avh
```

Instalador de Windows

Linux. Versións antigas: PPA

- `sudo add-apt-repository ppa:pdoes/gitflow-avh`
- `sudo apt-get update`
- `sudo apt-get install git-flow`

Instalación

Comprobación da versión instalada:

- `git flow version`

Subcomandos

\$ git flow

- **init**
- **feature**
- bugfix
- **release**
- **hotfix**
- support
- **version**
- config
- log



git flow init

```
$ git flow init help
```

```
$ git flow init
```

- Nome das ramas
- Prefixo das etiquetas

```
$ cat .git/config
```

```
$ git branch
```

```
$ git log
```

git flow init

```
$ git clone https://github.com/jquery/jquery.git
```

```
$ cd jquery
```

```
$ git branch -a -v
```

```
$ git flow init
```

```
$ git branch -a -v
```

```
$ cat .git/config
```

git flow feature

```
$ git flow feature help
```

```
$ git flow feature list
```

```
$ git flow feature start licenciaAdd
```

```
$ git branch
```

```
$ touch licencia.txt
```

```
$ echo 'The MIT License (MIT)' >> licencia.txt
```

```
$ echo 'Copyright (c) 2016 Jesús Amieiro Becerra' >>  
licencia.txt
```

```
$ git add licencia.txt
```



git flow feature (II)

```
$ git commit -m "Engado o arquivo de licencia"
```

```
$ git remote add origin
```

```
https://amieiro@bitbucket.org/amieiro/git\_flow.git
```

```
$ git push -u origin master
```

```
$ git push -u origin develop
```

```
$ git flow feature publish licenciaAdd
```

git flow feature (III)

```
# Usuario 2
```

```
$ git clone
```

```
https://JesusAmieiro@bitbucket.org/amieiro/git_flow  
.git
```

```
$ git branch -a -v
```

```
$ git flow init
```

```
$ git flow feature track licenciaAdd
```

```
$ git branch -a -v
```

git flow feature (IV)

```
# Usuario 1
```

```
$ git flow feature finish licenciaAdd
```

```
$ git branch -a -v
```

```
$ git push
```

```
$ git branch -a -v
```

git flow feature (V)

```
# Usuario 2
```

```
$ git checkout master; git pull origin master; git fetch  
--all -p; git branch -vv | grep ": gone]" | awk '{ print  
$1 }' | xargs -n 1 git branch -D
```

git flow release

```
$ git flow release help
```

```
$ git flow release list
```

```
$ git flow release start 0.1.0
```

```
$ git branch -a -v
```

```
$ echo "0.1.0" >> version.txt
```

```
$ git add version.txt
```

```
$ git commit -m "Engado o arquivo de versión ao  
proxecto"
```

```
$ git flow release publish 0.1.0
```



git flow release (II)

```
$ echo "Puedes consultar la licencia en la URL  
https://opensource.org/licenses/MIT" >> licencia.txt
```

```
$ git commit -am "Engado la URL de consulta no  
archivo de licencia"
```

```
$ git flow release publish 0.1.0
```

```
$ git flow release finish -F -p 0.1.0
```

git flow hotfix

```
$ git flow hotfix help
```

```
$ git flow hotfix list
```

```
$ git flow hotfix start 0.1.1
```

```
$ echo "0.1.1" > version.txt
```

```
$ echo "vendor/" >> .gitignore
```

```
$ git add .
```

```
$ git commit -m "Añado el archivo .gitignore"
```

```
$ git flow hotfix publish 0.1.1
```



git flow hotfix (II)

```
$ git branch -a -v
```

```
$ git flow hotfix finish --help
```

```
$ git flow hotfix finish -F -p 0.1.1
```

```
$ git branch -a -v
```

```
$ git log --oneline --all --decorate --graph
```

Onde continuar?

Libro "git-flow"

<http://nvie.com/posts/a-successful-git-branching-model/>

<https://github.com/petervanderdoes/gitflow-avh/wiki>

<https://yakiloo.com/getting-started-git-flow/>



Exercicio

Exercicio (I)

Por parellas: usuario1 (U1) e usuario2 (U2).

Crear 2 contas (1 por usuario) en Bitbucket ou sitio similar.

U1: Crear un proxecto con git-flow.

U1: Crear o arquivo de licenza MIT.

U2: Clonar o proxecto e inicializalo con git-flow.

U2: Crear o arquivo ".gitignore", excluindo do seguimento os arquivos "png."

Exercicio (II)

U1: Crear a release 0.1.0, cun arquivo "version.txt" onde se indique a versión e publicala.

U2: Crear un arquivo "index.html" con contido válido.

U2: Crear un arquivo "style.css" con contido válido.

U1: resolver un erro coa licencia en produción, que tiña que ser GPL v2 en vez de MIT.

U2. Crear a release 0.2.0, cos novos cambios, e publicalos.

www.jesusamieiro.com

