

Control de versiones con **git**



¿Para qué sirve un control de versiones?

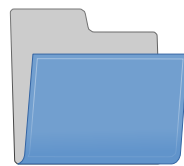
¿Para qué sirve?

¿Para qué sirve?

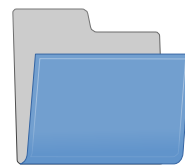
Para llevar registro de los cambios en archivos o proyectos

¿Para qué sirve?

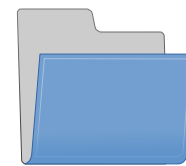
Para llevar registro de los cambios en archivos o proyectos



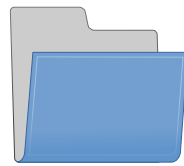
Tesis



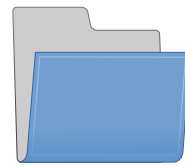
Tesis-hoy



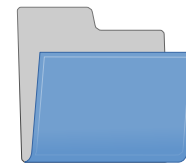
Tesis-final



Tesis-final
-final



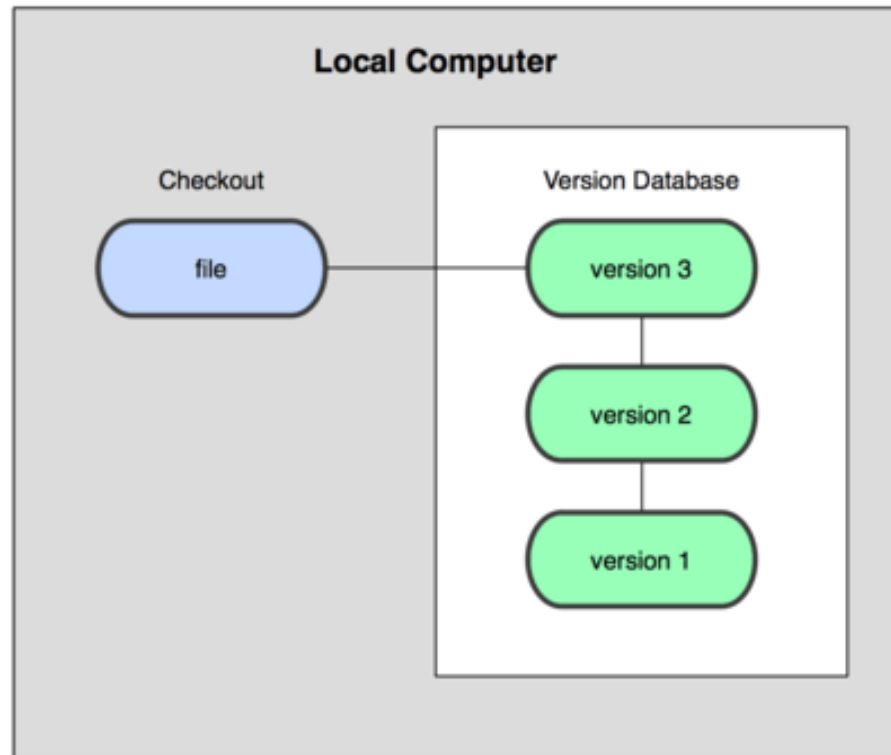
Tesis-si



Tesis-esta-no

Primer intento

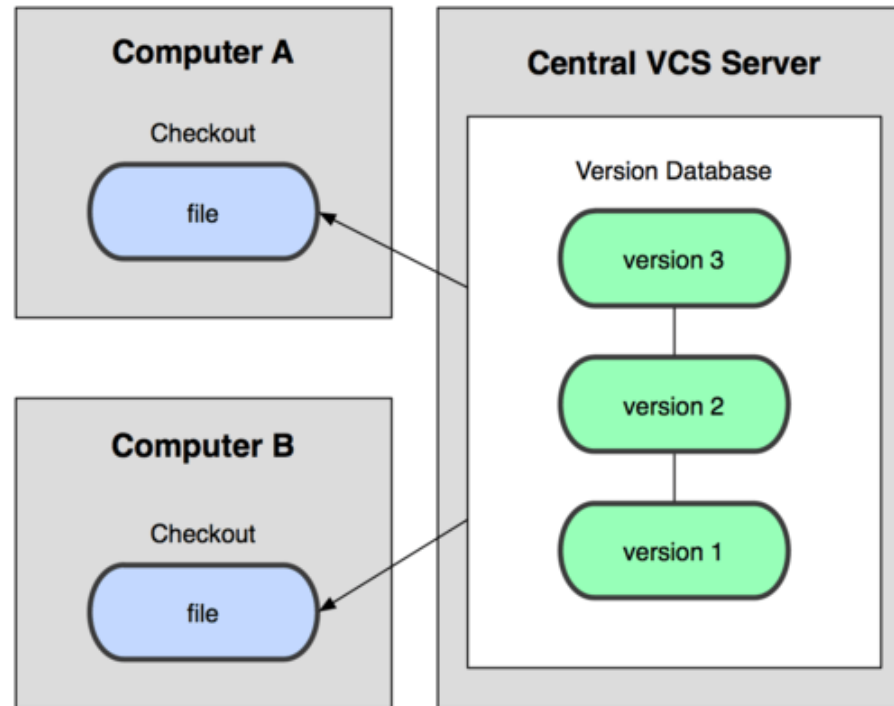
Primer intento



Ej.: RCS (Revision Control System)

Trabajo en equipo, pero centralizado

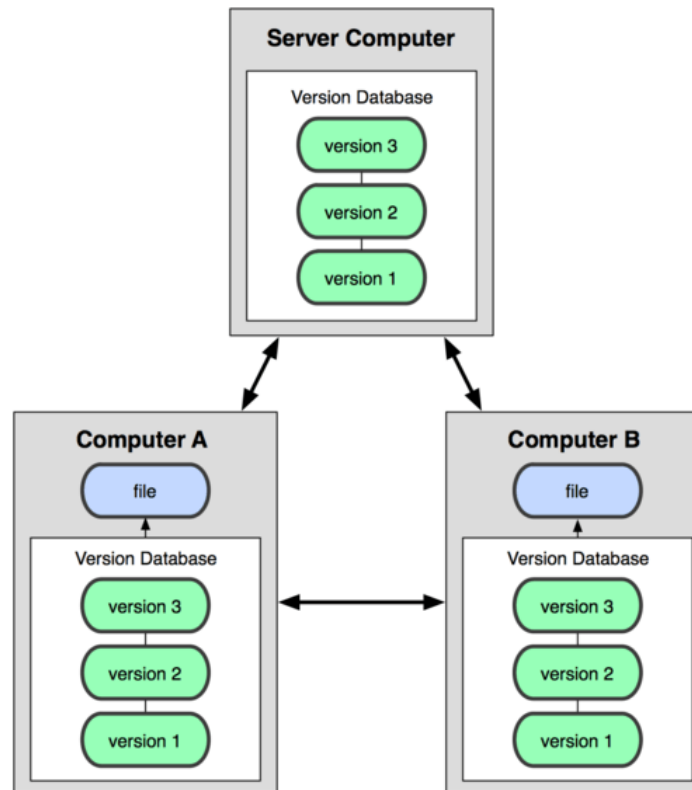
Trabajo en equipo, pero centralizado



Ej.: CVS, Subversion, etc.

Trabajo en equipo distribuido

Trabajo en equipo distribuido



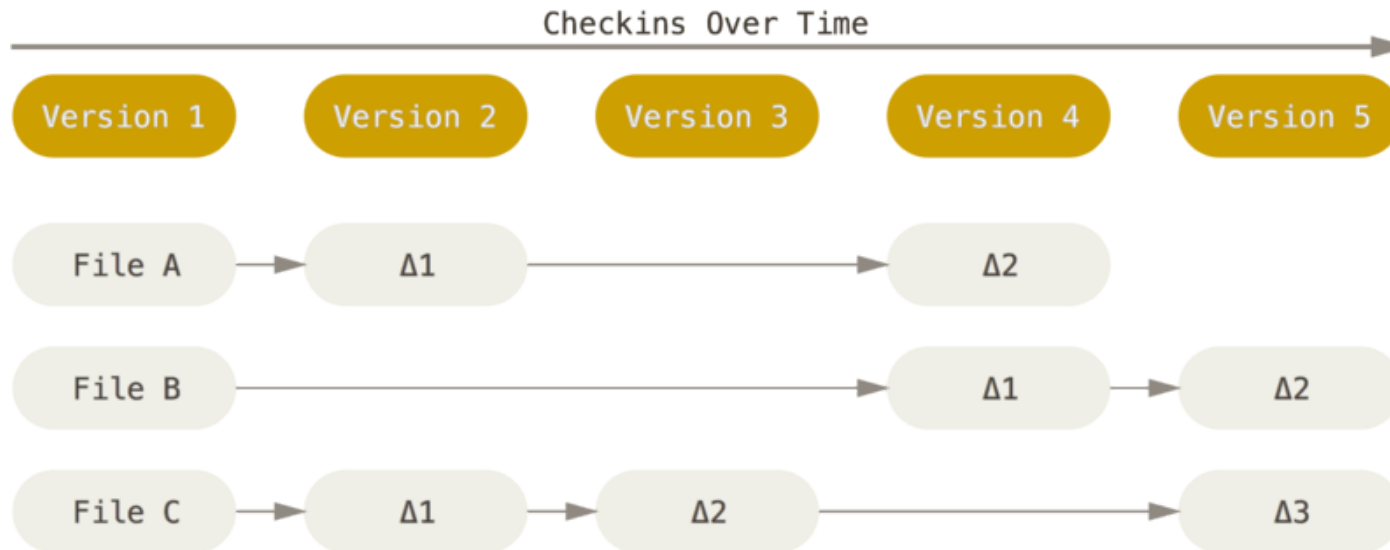
Ej.: git, Bazaar, etc.



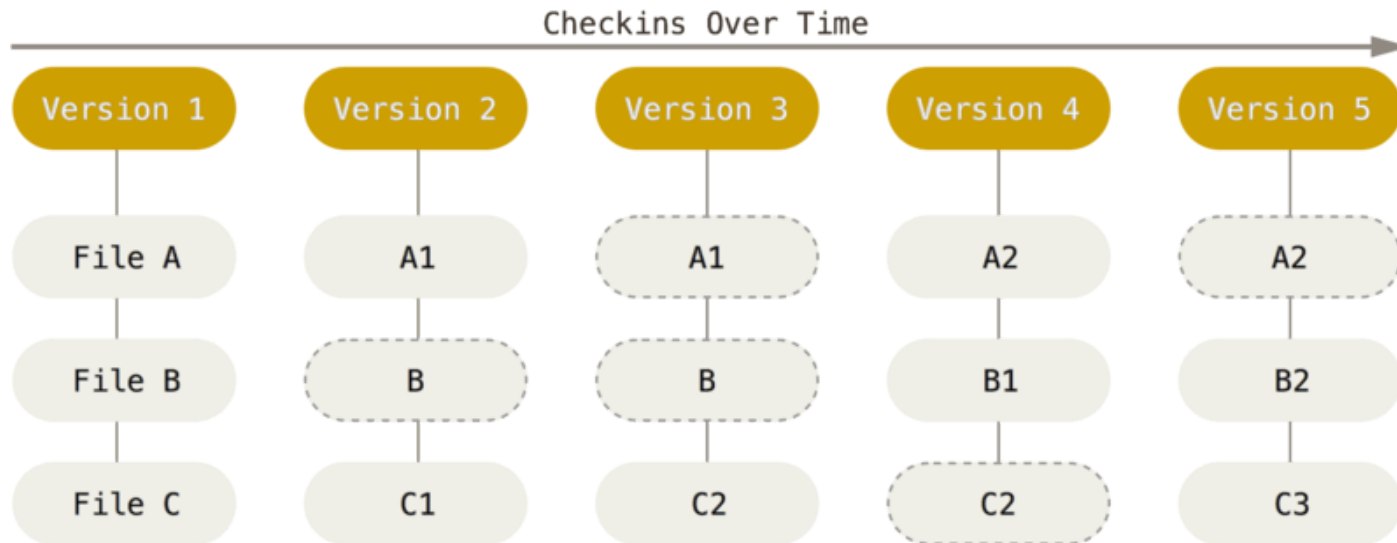
La diferencia de **git** frente a los otros

Los otros...

Los otros...

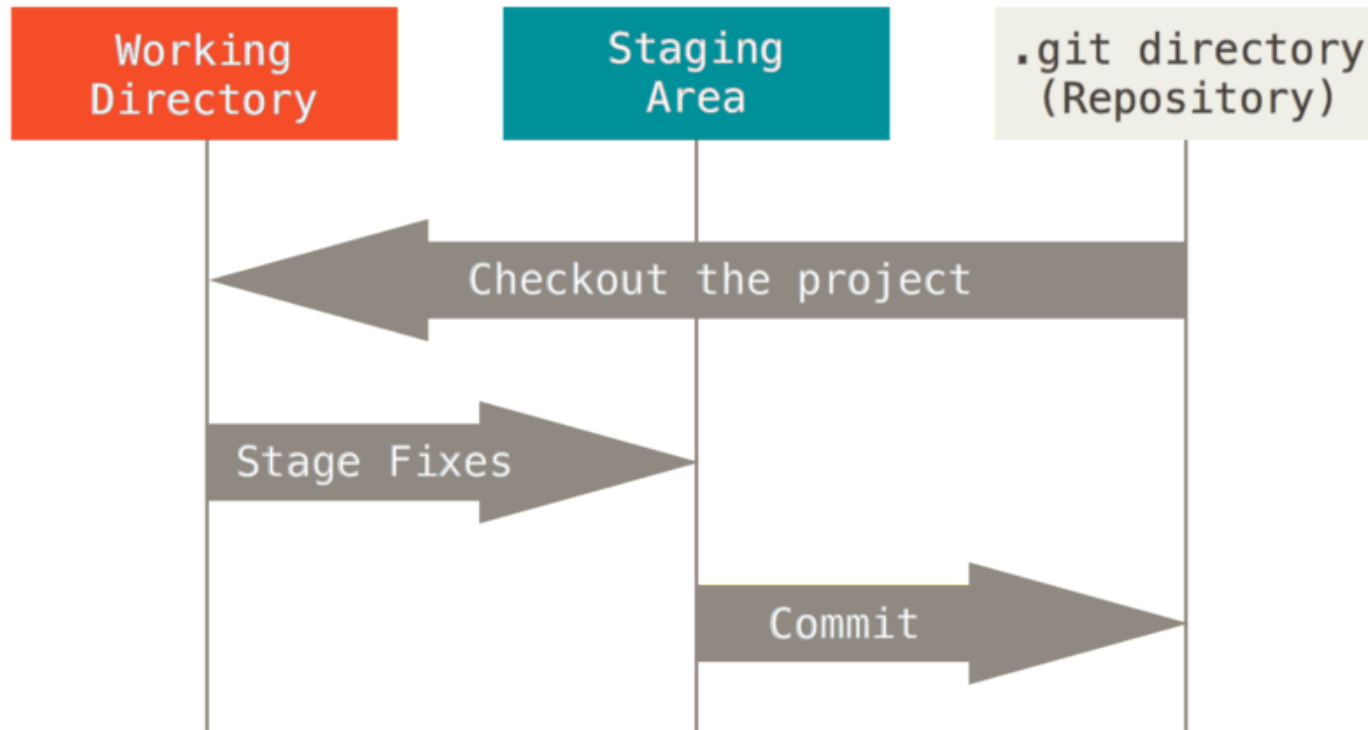


git



Los tres estados

Los tres estados





Instalando, configurando e iniciando un
control de versiones

Para Instalar

Para Instalar

en Debian y Ubuntu:

```
$ sudo apt install git
```

Para otras distribuciones:

<https://git-scm.com/download/>

Para Instalar

en Debian y Ubuntu:

```
$ sudo apt install git
```

Para otras distribuciones:

<https://git-scm.com/download/>

Para los merge es altamente recomendado el meld

```
$ sudo apt install meld
```


Para Configurar

Para Configurar

Porque siempre es bueno tener alguien a quien echarle la culpa...

Para Configurar

Porque siempre es bueno tener alguien a quien echarle la culpa...

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

Para Configurar

Porque siempre es bueno tener alguien a quien echarle la culpa...

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

Para editar los commits

```
$ git config --global core.editor /usr/bin/vim
```

Para Configurar

Porque siempre es bueno tener alguien a quien echarle la culpa...

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

Para editar los commits

```
$ git config --global core.editor /usr/bin/vim
```

Para hacer los merge a pata

```
$ git config --global merge.tool /usr/bin/meld
```

Para Configurar

Para Configurar

La configuración *global* del usuario se puede encontrar en
`~/.gitconfig`

Para Configurar

La configuración *global* del usuario se puede encontrar en
`~/.gitconfig`

La configuración *local* de cada proyecto en `.git/config`

Para Configurar

La configuración *global* del usuario se puede encontrar en
~/`.gitconfig`

La configuración *local* de cada proyecto en `.git/config`

La configuración *global* de todos los usuarios en
`/etc/gitconfig`

Para Configurar

La configuración *global* del usuario se puede encontrar en
~/`.gitconfig`

La configuración *local* de cada proyecto en `.git/config`

La configuración *global* de todos los usuarios en
`/etc/gitconfig`

Se puede chequear la lista de configuraciones con

Para Configurar

La configuración *global* del usuario se puede encontrar en
~/`.gitconfig`

La configuración *local* de cada proyecto en `.git/config`

La configuración *global* de todos los usuarios en
`/etc/gitconfig`

Se puede chequear la lista de configuraciones con

```
$ git config --list
```

Iniciando

Iniciando

Usando el repo de otro

Iniciando

Usando el repo de otro

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

Iniciando

Usando el repo de otro

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

Esto crea una carpeta llamada `git` y pone dentro todo el repo desde el repositorio remoto

Iniciando

Usando el repo de otro

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

Esto crea una carpeta llamada `git` y pone dentro todo el repo desde el repositorio remoto

O se le puede poner un nombre cualquiera

Iniciando

Usando el repo de otro

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

Esto crea una carpeta llamada `git` y pone dentro todo el repo desde el repositorio remoto

O se le puede poner un nombre cualquiera

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

Iniciando

Iniciando

O se puede arrancar el propio

Iniciando

O se puede arrancar el propio

```
$ git init propio
```

Iniciando

O se puede arrancar el propio

```
$ git init propio
```

Esto crea una carpeta `propio` y comienza el repo ahí (archivos de configuración, etc.)

Iniciando

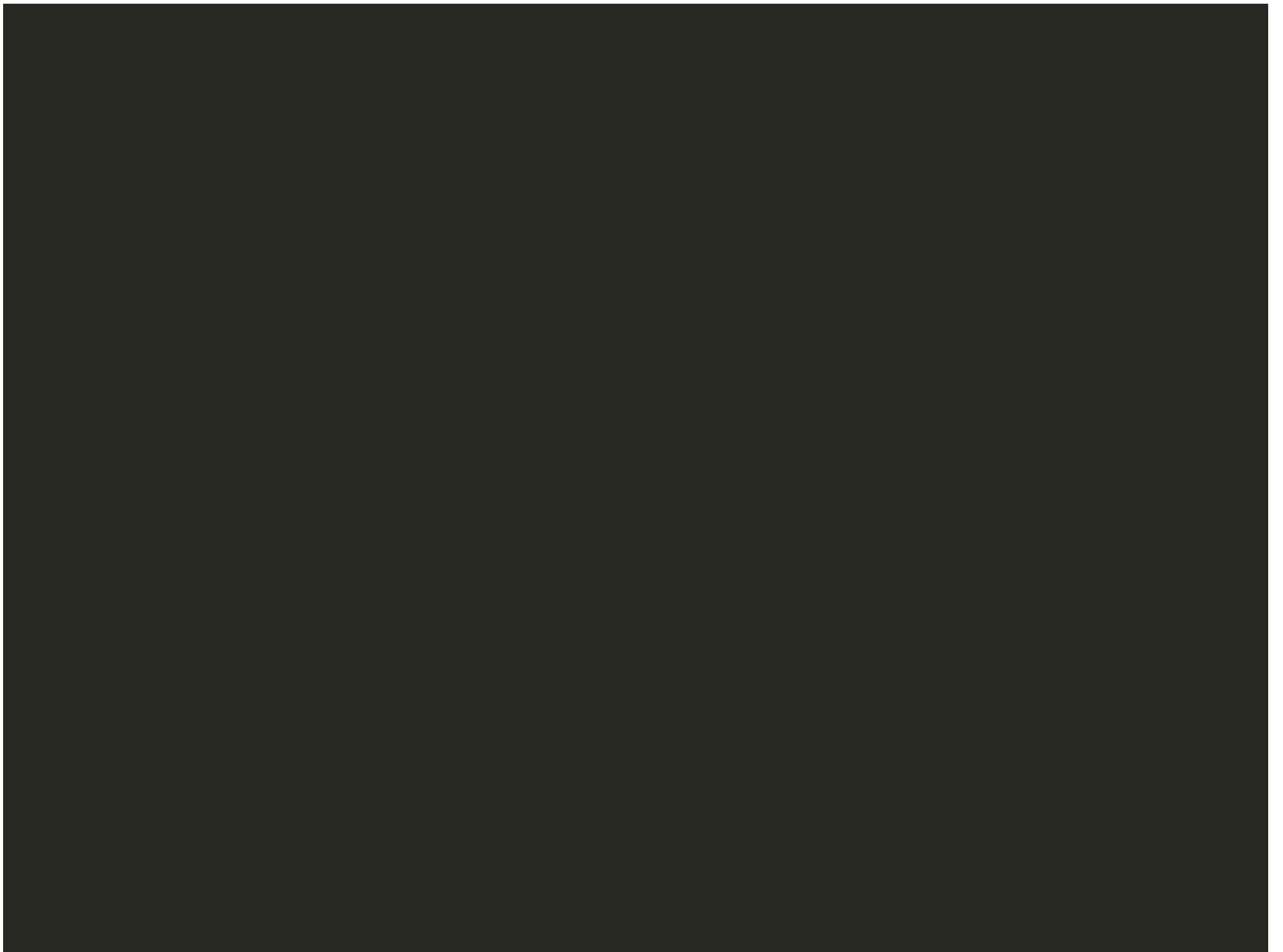
O se puede arrancar el propio

```
$ git init propio
```

Esto crea una carpeta `propio` y comienza el repo ahí (archivos de configuración, etc.)

...o lo mismo a pata

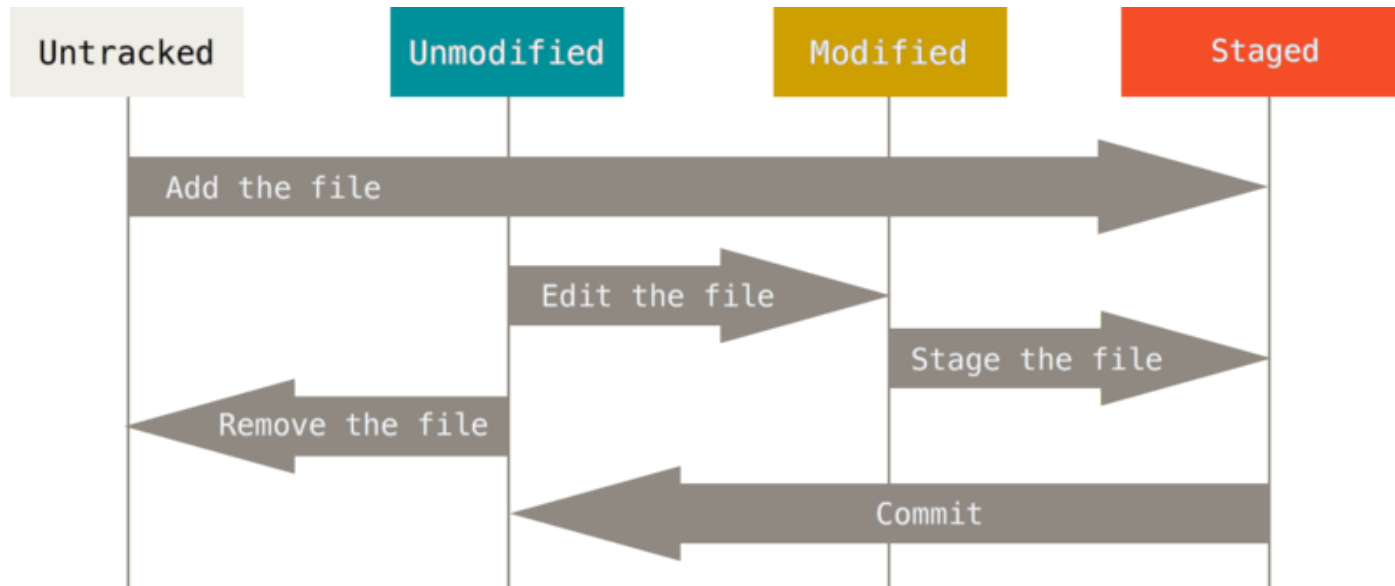
```
$ mkdir propio  
$ cd propio  
$ git init
```



Ahora si... usando **git**

El ciclo de vida de los archivos de git

El ciclo de vida de los archivos de git



Revisar el estado de los archivos

Revisar el estado de los archivos

```
$ cd propio  
$
```

Revisar el estado de los archivos

```
$ cd propio  
$ git status
```

Revisar el estado de los archivos

```
$ cd propio
$ git status
En la rama master

No hay commits todavía

no hay nada para confirmar (crea/copia archivos y usa
"git add" para hacerles seguimiento)
$
```

Revisar el estado de los archivos

```
$ cd propio
$ git status
En la rama master

No hay commits todavía

no hay nada para confirmar (crea/copia archivos y usa
"git add" para hacerles seguimiento)
$ touch README.md
$
```

Revisar el estado de los archivos

```
$ cd propio
$ git status
En la rama master

No hay commits todavía

no hay nada para confirmar (crea/copia archivos y usa
"git add" para hacerles seguimiento)
$ touch README.md
$ git status
```


Revisar el estado de los archivos

```
$ cd propio
$ git status
En la rama master

No hay commits todavía

no hay nada para confirmar (crea/copia archivos y usa
                             "git add" para hacerles seguimiento)
$ touch README.md
$ git status
En la rama master

No hay commits todavía

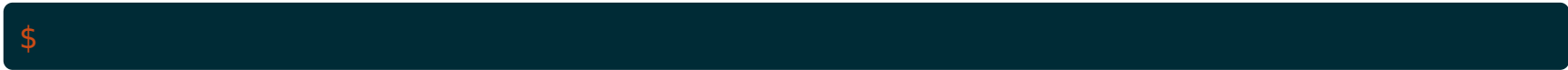
Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se
                               será confirmado)

  README.md

no hay nada agregado al commit pero hay archivos sin seguimiento
                             presentes (usa "git add" para hacerles seguimiento)
```

Rastrear archivos nuevos

Rastrear archivos nuevos



Rastrear archivos nuevos

```
$ git add README.md  
$
```

Rastrear archivos nuevos

```
$ git add README.md  
$ git status
```

Rastrear archivos nuevos

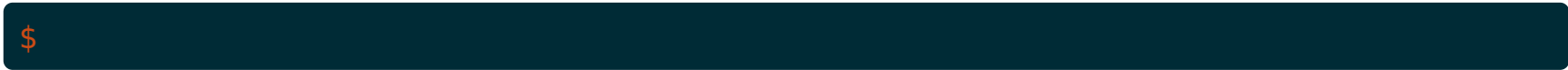
```
$ git add README.md
$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)

nuevo archivo: README.md
```

Preparar archivos modificados



Preparar archivos modificados

```
$ echo "Modificando README.md" > README.md  
$
```


Preparar archivos modificados

```
$ echo "Modificando README.md" > README.md  
$ git status
```

Preparar archivos modificados

```
$ echo "Modificando README.md" > README.md
$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)

nuevo archivo: README.md

Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git checkout -- <archivo>..." para descartar los cambios en
                                     el directorio de trabajo)

modificado: README.md
```

Método abreviado (de status)

Método abreviado (de status)

```
$ git status -s  
M README  
MM Rakefile  
A lib/git.rb  
M lib/simplegit.rb  
?? LICENSE.txt
```

Método abreviado (de status)

```
$ git status -s
M README
MM Rakefile
A lib/git.rb
M lib/simplegit.rb
?? LICENSE.txt
```

?? es un archivo que no está bajo seguimiento (untracked)

A es un archivo que está agregado y preparado para confirmación (staged)

M es un archivo que fue preparado alguna vez, pero fue modificado

Método abreviado (de status)

```
$ git status -s  
M README  
MM Rakefile  
A lib/git.rb  
M lib/simplegit.rb  
?? LICENSE.txt
```

Método abreviado (de status)

```
$ git status -s  
M README  
MM Rakefile  
A lib/git.rb  
M lib/simplegit.rb  
?? LICENSE.txt
```

Se pueden ver dos columnas, a la izquierda está el estado preparado para confirmar y a la derecha el estado si preparar

Ver los cambios preparados y no preparados

Ver los cambios preparados y no preparados

```
$ cat README.md  
Modificando README.md  
$
```

Ver los cambios preparados y no preparados

```
$ cat README.md  
Modificando README.md  
$ git status -s  
AM README.md  
$
```

Ver los cambios preparados y no preparados

```
$ cat README.md
Modificando README.md
$ git status -s
AM README.md
$ git diff
diff --git a/README.md b/README.md
index e69de29..a71e443 100644
--- a/README.md
+++ b/README.md
@@ -0,0 +1 @@
+Modificando README.md
$
```

Ver los cambios preparados y no preparados

```
$ cat README.md
Modificando README.md
$ git status -s
AM README.md
$ git diff
diff --git a/README.md b/README.md
index e69de29..a71e443 100644
--- a/README.md
+++ b/README.md
@@ -0,0 +1 @@
+Modificando README.md
$ git add README.md
$
```

Ver los cambios preparados y no preparados

```
$ cat README.md
Modificando README.md
$ git status -s
AM README.md
$ git diff
diff --git a/README.md b/README.md
index e69de29..a71e443 100644
--- a/README.md
+++ b/README.md
@@ -0,0 +1 @@
+Modificando README.md
$ git add README.md
$ git status -s
A README.md
$
```

Ver los cambios preparados y no preparados

```
$ cat README.md
Modificando README.md
$ git status -s
AM README.md
$ git diff
diff --git a/README.md b/README.md
index e69de29..a71e443 100644
--- a/README.md
+++ b/README.md
@@ -0,0 +1 @@
+Modificando README.md
$ git add README.md
$ git status -s
A README.md
$ git diff
$
```

Ver los cambios preparados y no preparados

Ver los cambios preparados y no preparados

`git diff` compara el directorio de trabajo con lo que esta preparado para confirmar (working directory vs staging area)

Ver los cambios preparados y no preparados

`git diff` compara el directorio de trabajo con lo que está preparado para confirmar (working directory vs staging area)

Para comparar lo preparado con lo último confirmado (o lo que está listo para confirmar) se usa

```
$ git diff --staged
```

ó

```
$ git diff --cached
```

Confirmar los cambios

Confirmar los cambios

```
$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)

nuevo archivo: README.md
$
```

Confirmar los cambios

```
$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)

nuevo archivo: README.md
$ git commit
```

Confirmar los cambios

```
$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)

nuevo archivo: README.md
$ git commit
```

En este momento se abre el editor que se haya configurado con `git config --global core.editor` o en su defecto con lo que tenga la variable de entorno `$EDITOR`

Confirmar los cambios (editor)

Confirmar los cambios (editor)

```
# Por favor ingrese el mensaje del commit para sus cambios. Las líneas que comiencen
# con '#' serán ignoradas, y un mensaje vacío aborta el commit.
#
# En la rama master
#
# Confirmación inicial
#
# Cambios a ser confirmados:
#   nuevo archivo: README.md
#
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Confirmar los cambios (línea de comandos)

Confirmar los cambios (línea de comandos)

```
$ git commit -m "Agrega README.md"
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$ echo "Otra línea" >> README.md
$
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$ echo "Otra línea" >> README.md
$ git status -s
M README.md
$
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$ echo "Otra línea" >> README.md
$ git status -s
M README.md
$ git add README.md
$
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$ echo "Otra línea" >> README.md
$ git status -s
M README.md
$ git add README.md
$ git commit -m "Agrega una nueva línea a README.md"
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$ echo "Otra línea" >> README.md
$ git status -s
M README.md
$
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$ echo "Otra línea" >> README.md
$ git status -s
M README.md
$ git commit -a -m "Agrega una nueva línea a README.md"
```


Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$ echo "Otra línea" >> README.md
$ git status -s
M README.md
$ git commit -a -m "Agrega una nueva línea a README.md"
[master 1090bdb] Agrega una nueva línea a README.md
1 file changed, 1 insertion(+)
$
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
$ echo "Otra línea" >> README.md
$ git status -s
M README.md
$ git commit -a -m "Agrega una nueva línea a README.md"
[master 1090bdb] Agrega una nueva línea a README.md
1 file changed, 1 insertion(+)
$ git status
```

Confirmar los cambios

```
$ git commit -m "Agrega README.md"
[master (commit-raíz) dallada] Agrega README.md
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
$ echo "Otra línea" >> README.md
$ git status -s
M README.md
$ git commit -a -m "Agrega una nueva línea a README.md"
[master 1090bdb] Agrega una nueva línea a README.md
 1 file changed, 1 insertion(+)
$ git status
En la rama master
nada para hacer commit, el árbol de trabajo está limpio
$
```

Eliminar archivos

Eliminar archivos (sin querer)

Eliminar archivos (sin querer)

```
$ rm README.md
```

Eliminar archivos (sin querer)

```
$ rm README.md  
$ git status -s
```

Eliminar archivos (sin querer)

```
$ rm README.md  
$ git status -s  
D README.md  
$
```


Eliminar archivos (sin querer)

```
$ rm README.md  
$ git status -s  
D README.md  
$ git add .  
$
```

Eliminar archivos (sin querer)

```
$ rm README.md
$ git status -s
D README.md
$ git add .
$ git status -s
D README.md
$
```

Eliminar archivos (sin querer)

```
$ rm README.md
$ git status -s
D README.md
$ git add .
$ git status -s
D README.md
$ git reset HEAD README.md
$
```

Eliminar archivos (sin querer)

```
$ rm README.md
$ git status -s
D README.md
$ git add .
$ git status -s
D README.md
$ git reset HEAD README.md
$ ls
$
```

Eliminar archivos (sin querer)

```
$ rm README.md
$ git status -s
D README.md
$ git add .
$ git status -s
D README.md
$ git reset HEAD README.md
$ ls
$ git status -s
D README.md
$
```

Eliminar archivos (sin querer)

```
$ rm README.md
$ git status -s
D README.md
$ git add .
$ git status -s
D README.md
$ git reset HEAD README.md
$ ls
$ git status -s
D README.md
$ git checkout -- README.md
$
```

Eliminar archivos (sin querer)

```
$ rm README.md
$ git status -s
D README.md
$ git add .
$ git status -s
D README.md
$ git reset HEAD README.md
$ ls
$ git status -s
D README.md
$ git checkout -- README.md
$ ls
README.md
$
```

Eliminar archivos (queriendo)

Eliminar archivos (queriendo)

```
$ git rm README.md  
$
```

Eliminar archivos (queriendo)

```
$ git rm README.md  
$ git status -s  
D README.md  
$
```

Corregir un commit

Corregir un commit

Si queremos corregir el mensaje de un commit

Corregir un commit

Si queremos corregir el mensaje de un commit
o lo más común, se olvidó preparar un archivo...

Corregir un commit

Si queremos corregir el mensaje de un commit
o lo más común, se olvidó preparar un archivo...

```
$ git status -s  
M README.md  
?? LICENSE  
$
```

Corregir un commit

Si queremos corregir el mensaje de un commit o lo más común, se olvidó preparar un archivo...

```
$ git status -s
M README.md
?? LICENSE
$ git commit -a -m "Agrega archivo de licencia"
[master a8cae20] Agrega archivo de licencia
1 file changed, 1 insertion(+), 1 deletion(-)
$
```

Corregir un commit

Si queremos corregir el mensaje de un commit o lo más común, se olvidó preparar un archivo...

```
$ git status -s
M README.md
?? LICENSE
$ git commit -a -m "Agrega archivo de licencia"
[master a8cae20] Agrega archivo de licencia
1 file changed, 1 insertion(+), 1 deletion(-)
$ git status -s
?? LICENSE
$
```


Corregir un commit

Si queremos corregir el mensaje de un commit o lo más común, se olvidó preparar un archivo...

```
$ git status -s
M README.md
?? LICENSE
$ git commit -a -m "Agrega archivo de licencia"
[master a8cae20] Agrega archivo de licencia
1 file changed, 1 insertion(+), 1 deletion(-)
$ git status -s
?? LICENSE
$ git add LICENSE
$
```

Corregir un commit

Si queremos corregir el mensaje de un commit o lo más común, se olvidó preparar un archivo...

```
$ git status -s
M README.md
?? LICENSE
$ git commit -a -m "Agrega archivo de licencia"
[master a8cae20] Agrega archivo de licencia
1 file changed, 1 insertion(+), 1 deletion(-)
$ git status -s
?? LICENSE
$ git add LICENSE
$ git commit --amend
```

Corregir un commit

Etiquetado

Etiquetado

Las **etiquetas ligeras** son simplemente un puntero a un `commit` específico.

Etiquetado

Las **etiquetas ligeras** son simplemente un puntero a un `commit` específico.

Las **etiquetas anotadas** además contienen el nombre del etiquetador, correo electrónico y fecha; tienen un mensaje asociado; y pueden ser firmadas y verificadas con GNU Privacy Guard (GPG).

Etiquetado (ligero)

Etiquetado (ligero)

```
$ gitk
```

Etiquetado (ligero)

```
$ gitk
```

Archivo Editar Vista Ayuda

● **master** Agrega archivo de licencia
● Agrega una nueva línea a README.md
● Agrega README.md

Etiquetado (ligero)

```
$ gitk  
$
```

Etiquetado (ligero)

```
$ gitk  
$ git tag v1.0  
$
```

Etiquetado (ligero)

```
$ gitk  
$ git tag v1.0  
$ gitk
```

Etiquetado (ligero)

```
$ gitk  
$ git tag v1.0  
$ gitk
```

Archivo Editar Vista Ayuda

A Git commit history visualization. It shows a vertical line of three commits. The top commit is a yellow circle with a yellow arrow pointing to a yellow box labeled 'v1.0'. The middle commit is a blue circle with a blue arrow pointing to a green box labeled 'master'. The bottom commit is a blue circle. To the right of the top commit is the text 'Agrega archivo de licencia'. To the right of the middle commit is the text 'Agrega una nueva línea a README.md'. To the right of the bottom commit is the text 'Agrega README.md'.

● v1.0 → master Agrega archivo de licencia
● Agrega una nueva línea a README.md
● Agrega README.md

Etiquetado (ligero)

```
$ gitk  
$ git tag v1.0  
$ gitk  
$
```

Etiquetado (ligero)

```
$ gitk  
$ git tag v1.0  
$ gitk  
$ git show
```


Etiquetado (ligero)

```
$ gitk  
$ git tag v1.0  
$ gitk  
$ git show  
commit d1c1e65eb7de59cf654a...(HEAD -> master, tag: v1.0)  
Author: claudiojpaz <claudiojpaz@gmail.com>  
Date:   Wed Jul 3 23:37:31 2019 -0300
```

Agrega archivo de licencia

```
diff --git a/LICENSE b/LICENSE  
new file mode 100644  
index 0000000..e69de29
```

Etiquetado (anotado)

Etiquetado (anotado)

```
$ git tag --list  
v1.0  
$
```

Etiquetado (anotado)

```
$ git tag --list  
v1.0  
$ git tag -d v1.0  
Etiqueta 'v1.0' eliminada  
$
```

Etiquetado (anotado)

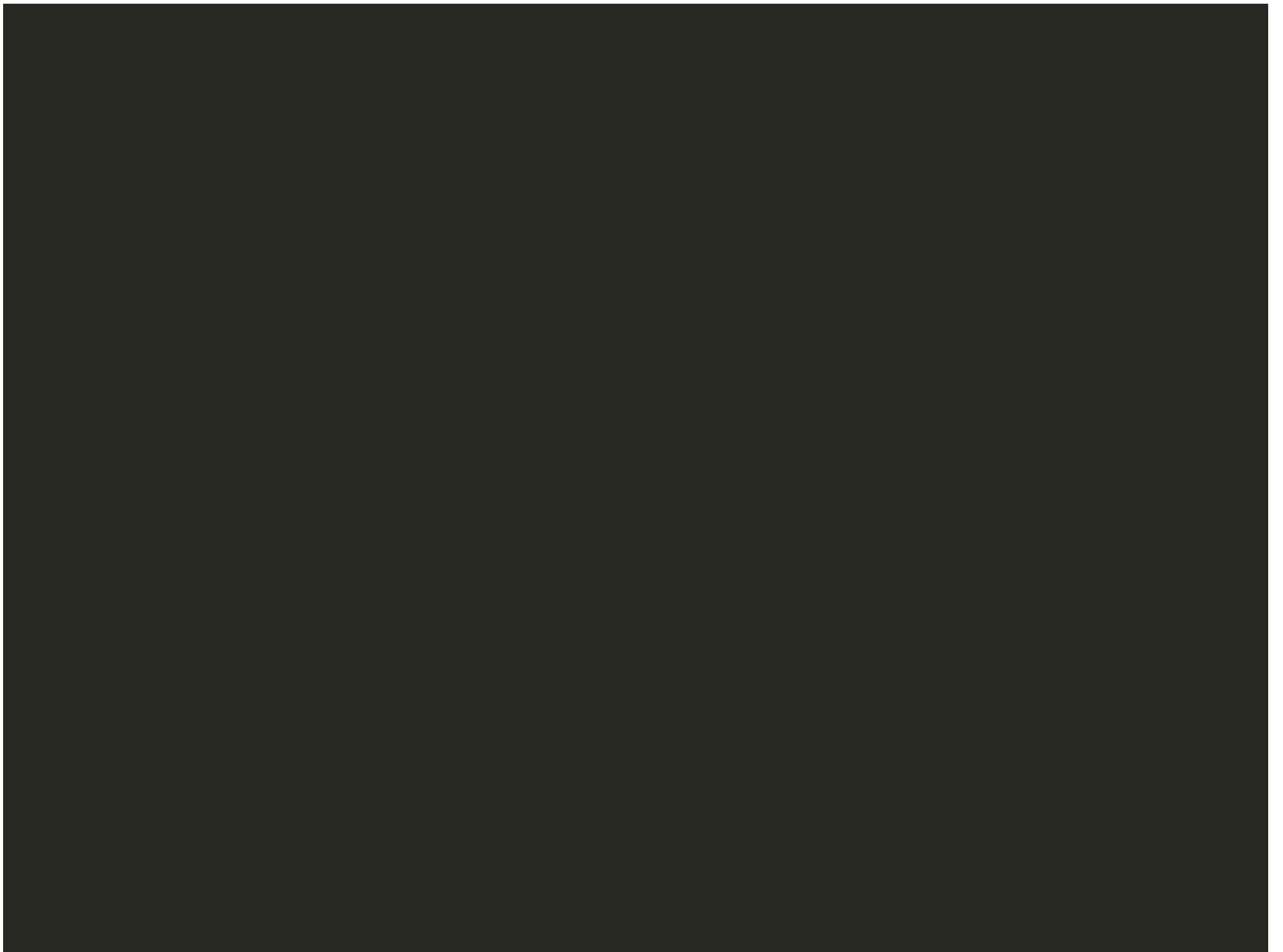
```
$ git tag --list  
v1.0  
$ git tag -d v1.0  
Etiqueta 'v1.0' eliminada  
$ git tag -a v1.0 -m "Primera etiqueta"  
$
```

Etiquetado (anotado)

```
$ git tag --list
v1.0
$ git tag -d v1.0
Etiqueta 'v1.0' eliminada
$ git tag -a v1.0 -m "Primera etiqueta"
$ git show v1.0
tag v1.0
Tagger: claudiojpaz <claudiojpaz@gmail.com>
Date: Thu Jul 4 00:27:21 2019 -0300

Primera etiqueta

commit d1c1e65eb7de59cf654a...(HEAD -> master, tag: v1.0)
...
```



¿Y las ramas?

Hasta ahora solo master

Hasta ahora solo master

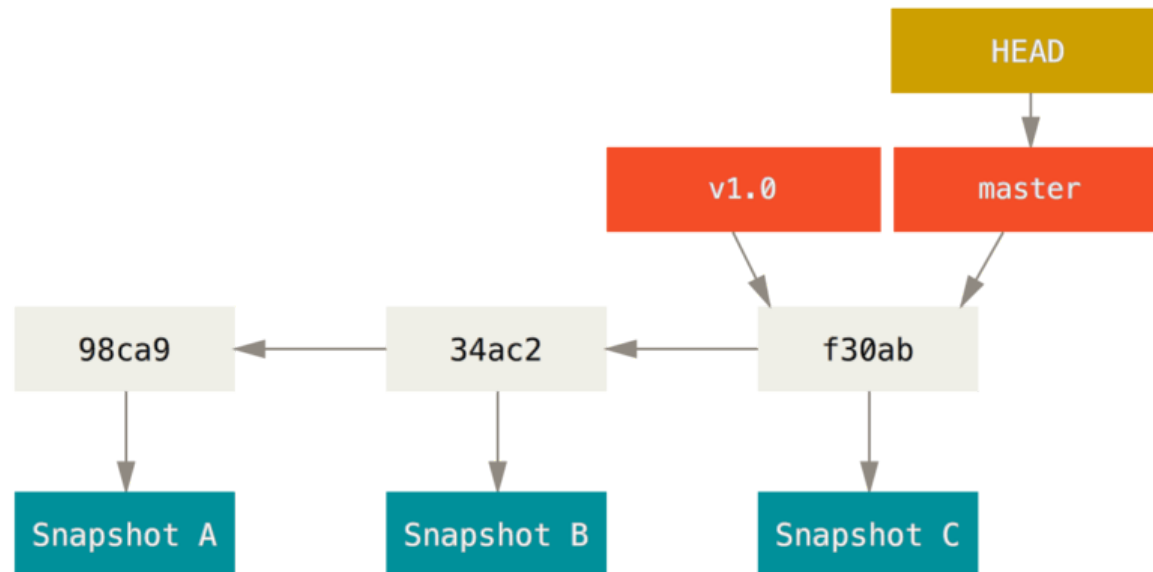
```
$ git log --oneline
```

Hasta ahora solo master

```
$ git log --oneline
f30ab (HEAD -> master, tag: v1.0) Agrega archivo de licencia
34ac2 Agrega una nueva línea a README.md
98ca9 Agrega README.md
```

Hasta ahora solo master

```
$ git log --oneline
f30ab (HEAD -> master, tag: v1.0) Agrega archivo de licencia
34ac2 Agrega una nueva línea a README.md
98ca9 Agrega README.md
```



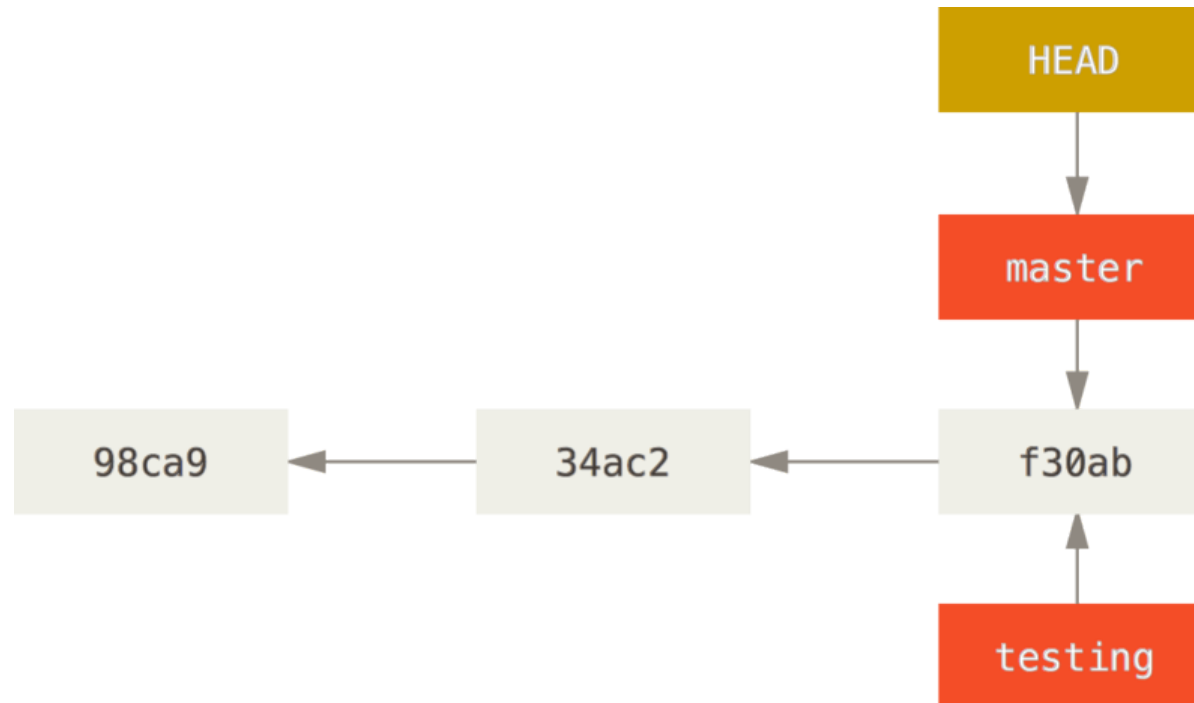
Creando una nueva rama

Creando una nueva rama

```
$ git branch testing  
$
```

Creando una nueva rama

```
$ git branch testing  
$
```



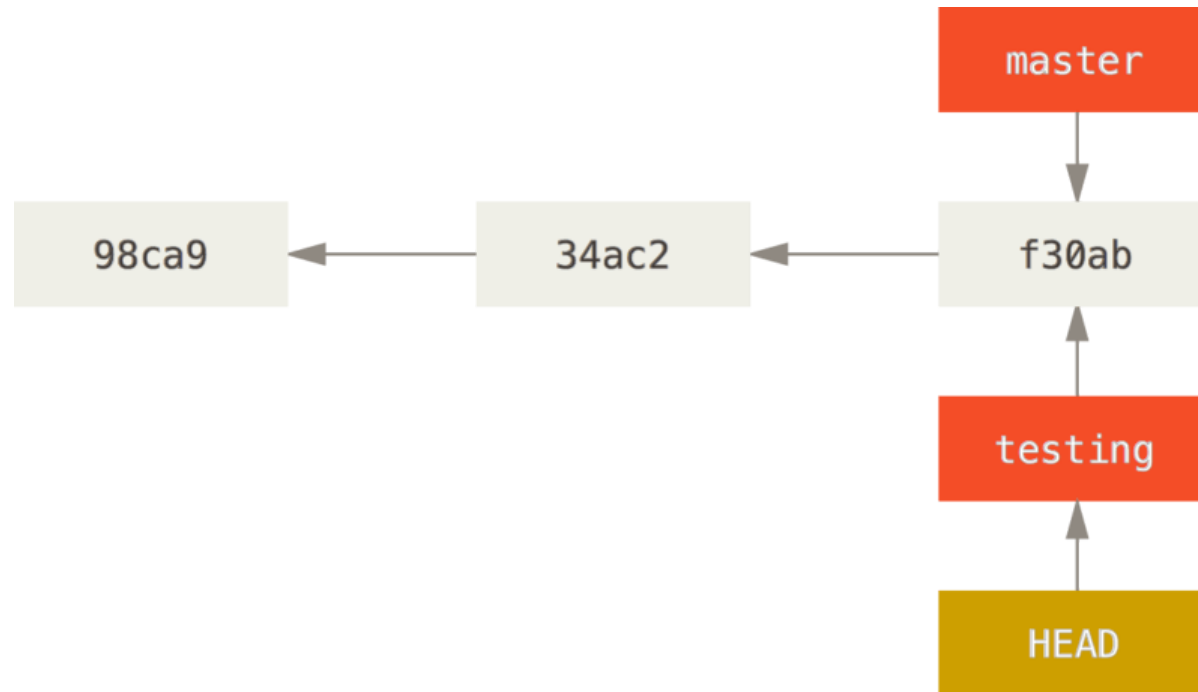
Cambiando de rama

Cambiando de rama

```
$ git checkout testing  
Cambiado a rama 'testing'  
$
```

Cambiando de rama

```
$ git checkout testing
Cambiado a rama 'testing'
$
```



Cambiando de rama

Cambiando de rama

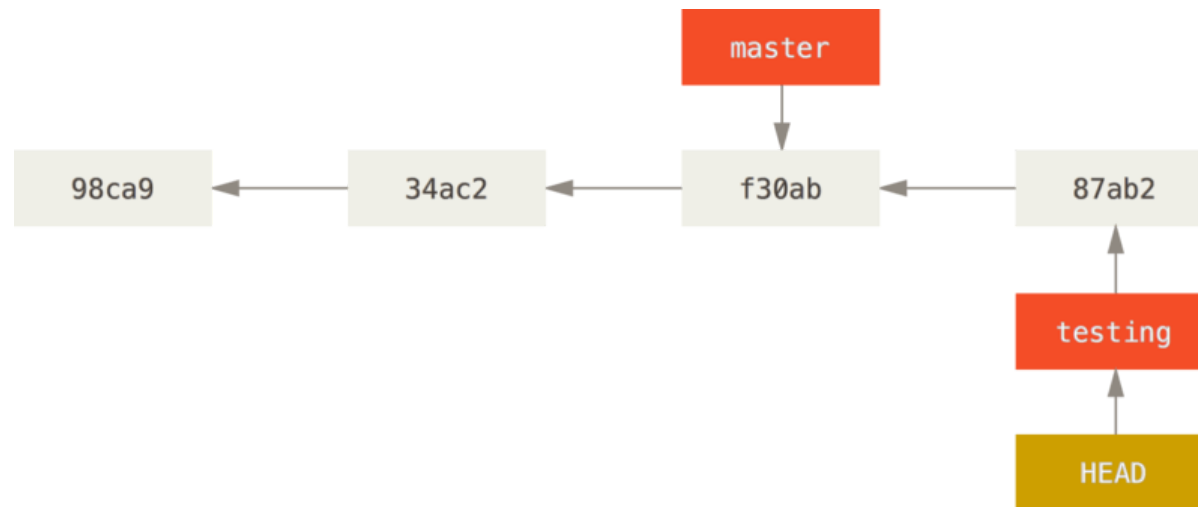
```
$ echo "Algo de GPL" >> LICENSE  
$
```

Cambiando de rama

```
$ echo "Algo de GPL" >> LICENSE
$ git commit -a -m "Agrega texto de licencia"
[testing 87ab2] Agrega texto de licencia
1 file changed, 1 insertion(+)
$
```

Cambiando de rama

```
$ echo "Algo de GPL" >> LICENSE
$ git commit -a -m "Agrega texto de licencia"
[testing 87ab2] Agrega texto de licencia
1 file changed, 1 insertion(+)
$
```



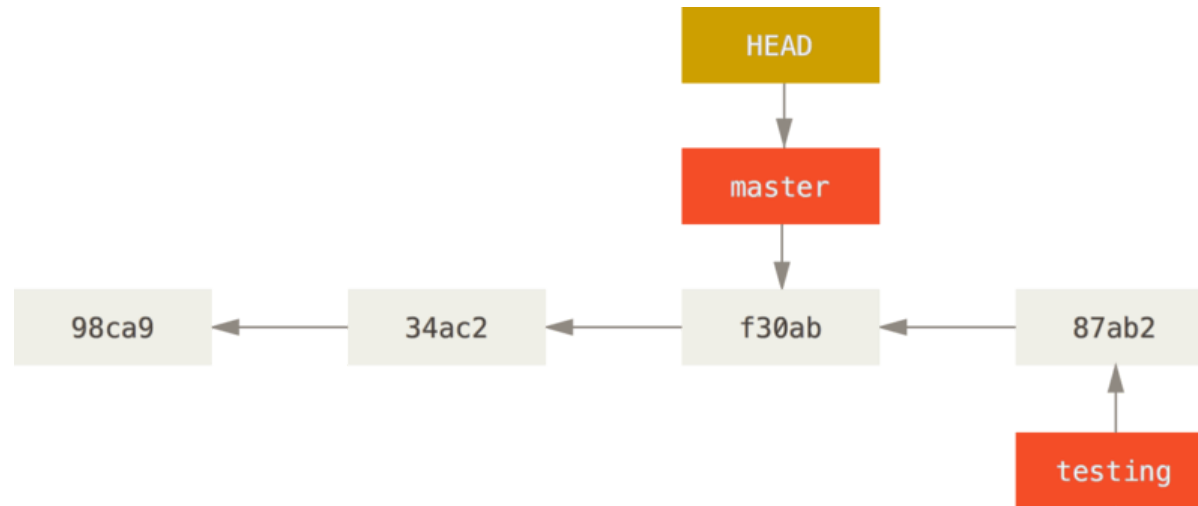
Cambiando de rama

Cambiando de rama

```
$ git checkout master  
Cambiado a rama 'master'  
$
```


Cambiando de rama

```
$ git checkout master  
Cambiado a rama 'master'  
$
```



Cambiando de rama

Cambiando de rama

```
$ echo "Algo más de GPL" >> LICENSE  
$
```

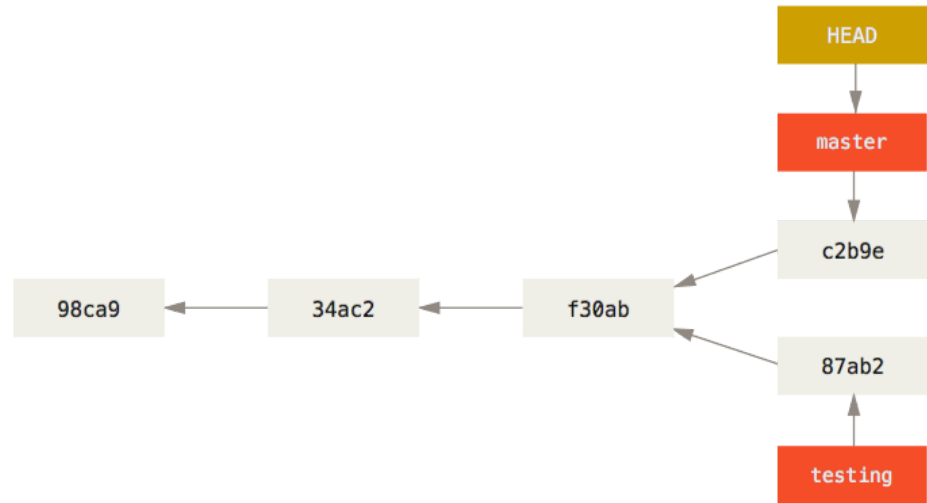
Cambiando de rama

Cambiando de rama

```
$ git commit -a -m "Agrega más texto a licencia"  
[master c2b9e] Agrega más texto a licencia  
1 file changed, 1 insertion(+)  
$
```

Cambiando de rama

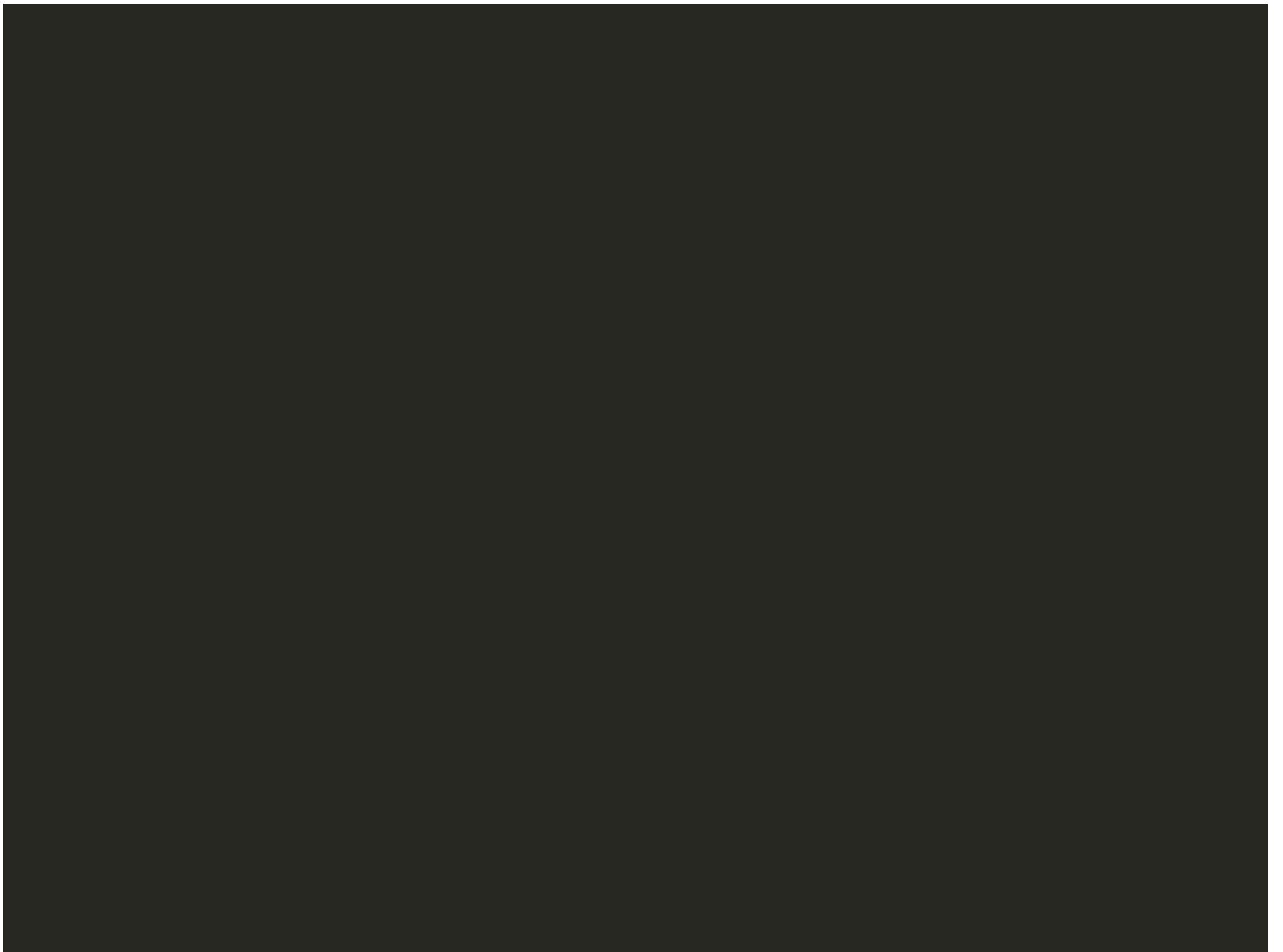
```
$ git commit -a -m "Agrega más texto a licencia"  
[master c2b9e] Agrega más texto a licencia  
1 file changed, 1 insertion(+)  
$
```



Cambiando de rama

Cambiando de rama

```
$ git log --oneline --graph --all
* c2b9e (HEAD -> master) Agrega más texto a licencia
| * 87ab2 (testing) Agrega texto de licencia
|/
* f30ab (tag: v1.0) Agrega archivo de licencia
* 34ac2 Agrega una nueva línea a README.md
* 98ca9 Agrega README.md
```

Fusionar (merge) Vs Reorganizar (rebase)

Procedimientos básicos de ramificación

Procedimientos básicos de ramificación



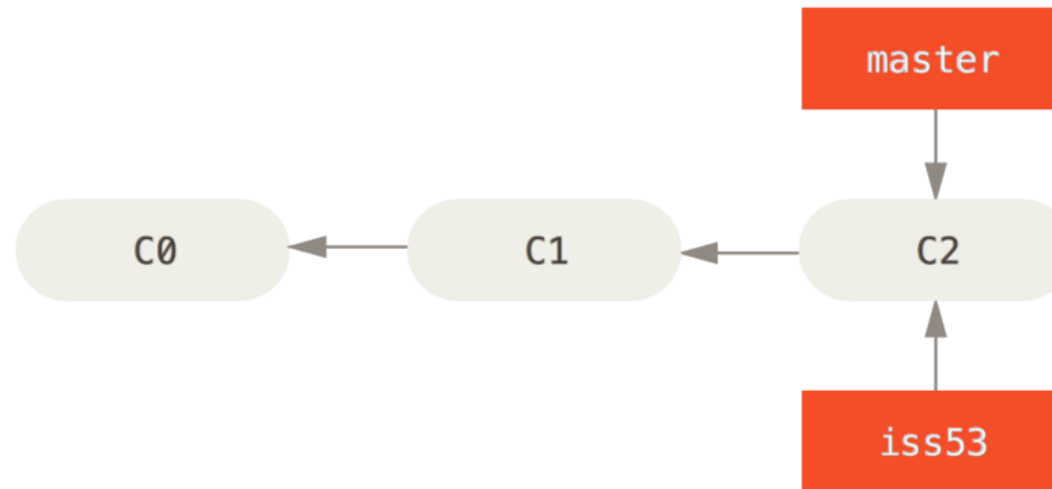
Procedimientos básicos de ramificación

Procedimientos básicos de ramificación

```
$ git checkout -b iss53
```

Procedimientos básicos de ramificación

```
$ git checkout -b iss53
```



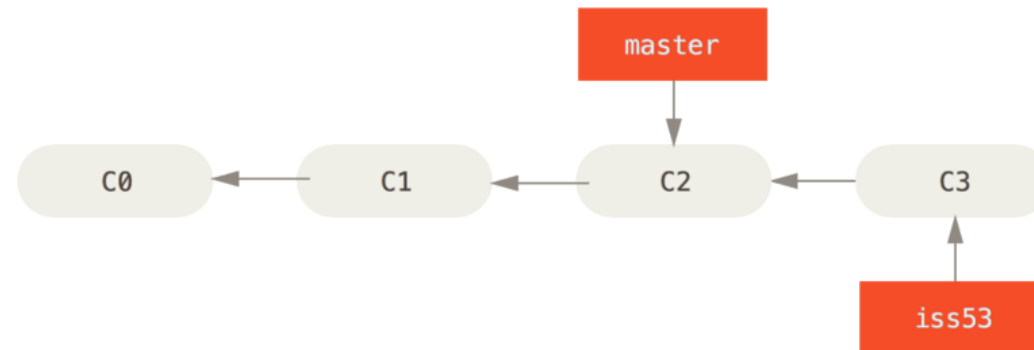
Procedimientos básicos de ramificación

Procedimientos básicos de ramificación

```
$ git commit -a -m "[iss53] Arregla algunas cosas"
```

Procedimientos básicos de ramificación

```
$ git commit -a -m "[iss53] Arregla algunas cosas"
```



Procedimientos básicos de ramificación

Procedimientos básicos de ramificación

```
$ git checkout master  
Cambiado a rama 'master'  
$
```

Procedimientos básicos de ramificación

```
$ git checkout master  
Cambiado a rama 'master'  
$ git checkout -b hotfix  
Cambiado a nueva rama 'hotfix'
```

Procedimientos básicos de ramificación

```
$ git checkout master  
Cambiado a rama 'master'  
$ git checkout -b hotfix  
Cambiado a nueva rama 'hotfix'  
  
...  
$
```

Procedimientos básicos de ramificación

```
$ git checkout master
Cambiado a rama 'master'
$ git checkout -b hotfix
Cambiado a nueva rama 'hotfix'

..
$ git commit -a -m "[hotfix] Arregla algo urgente"
```


Procedimientos básicos de ramificación

Procedimientos básicos de ramificación

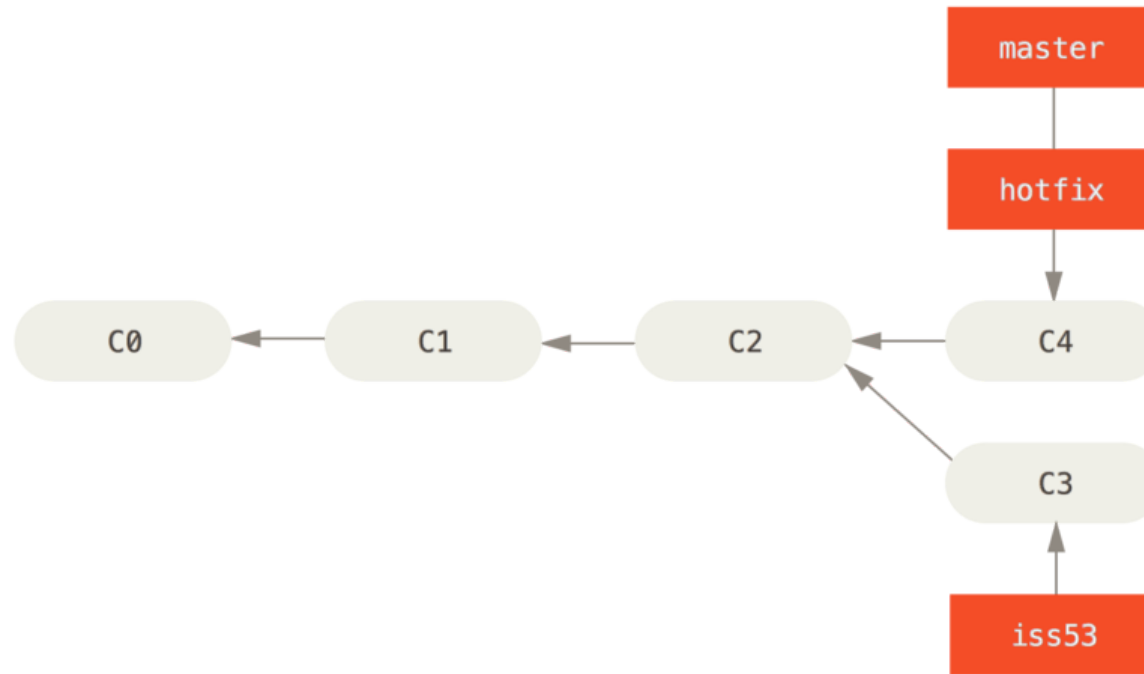
```
$ git checkout master  
Cambiado a rama 'master'
```

Procedimientos básicos de ramificación

```
$ git checkout master
Cambiado a rama 'master'
$ git merge hotfix
Actualizando 09277d1..54c0abe
Fast-forward
 otro_archivo | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 otro_archivo
```

Procedimientos básicos de ramificación

Procedimientos básicos de ramificación



Procedimientos básicos de ramificación

Procedimientos básicos de ramificación

```
$ git branch -d hotfix  
Eliminada la rama hotfix (era 54c0abe)..  
$
```

Procedimientos básicos de ramificación

Procedimientos básicos de ramificación

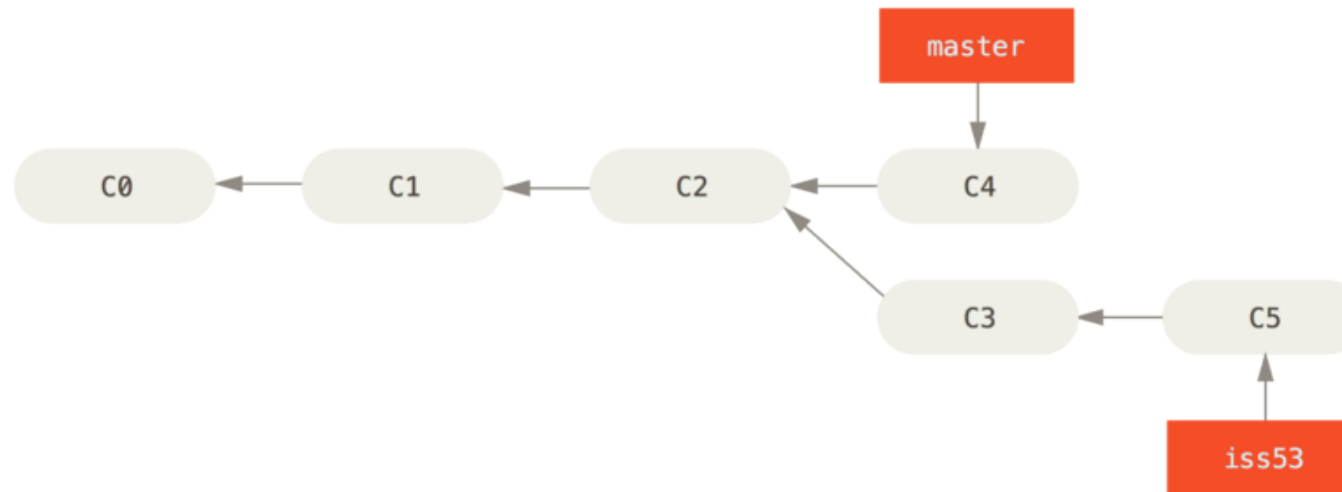
```
$ git checkout iss53
Cambiado a rama 'iss53'

...

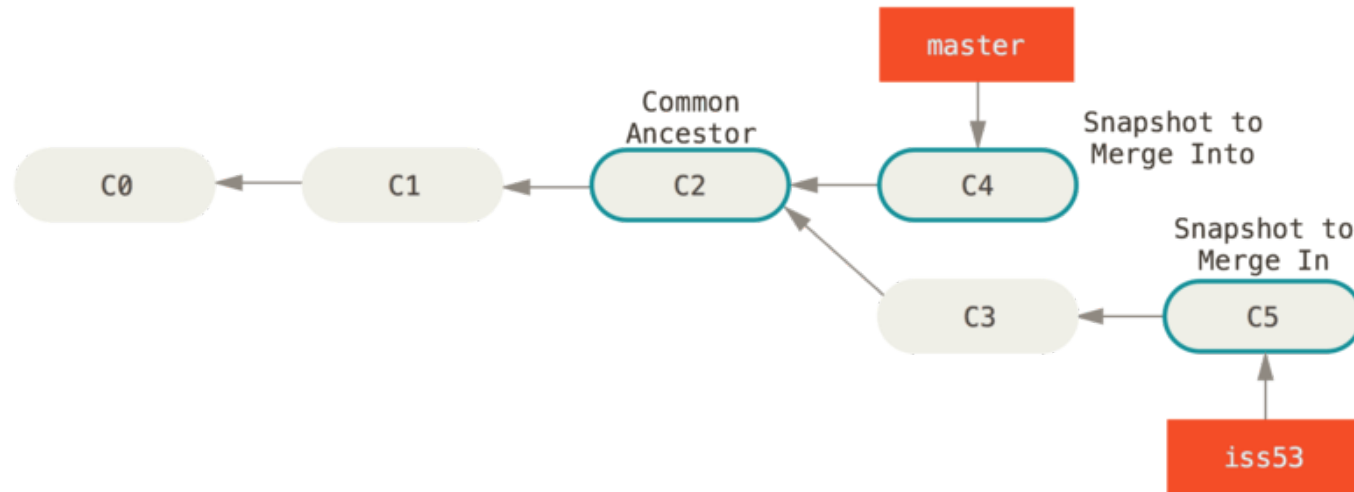
$ git commit -a -m "[iss53] Corrige todos los problemas"
[iss53 ad82d7a] [iss53] Corrige todos los problemas
1 file changed, 1 insertion(+)
```

Procedimientos básicos de ramificación

Procedimientos básicos de ramificación



merge



merge

merge

```
$ git checkout master  
Cambiado a rama 'master'  
$ git merge iss53
```

merge

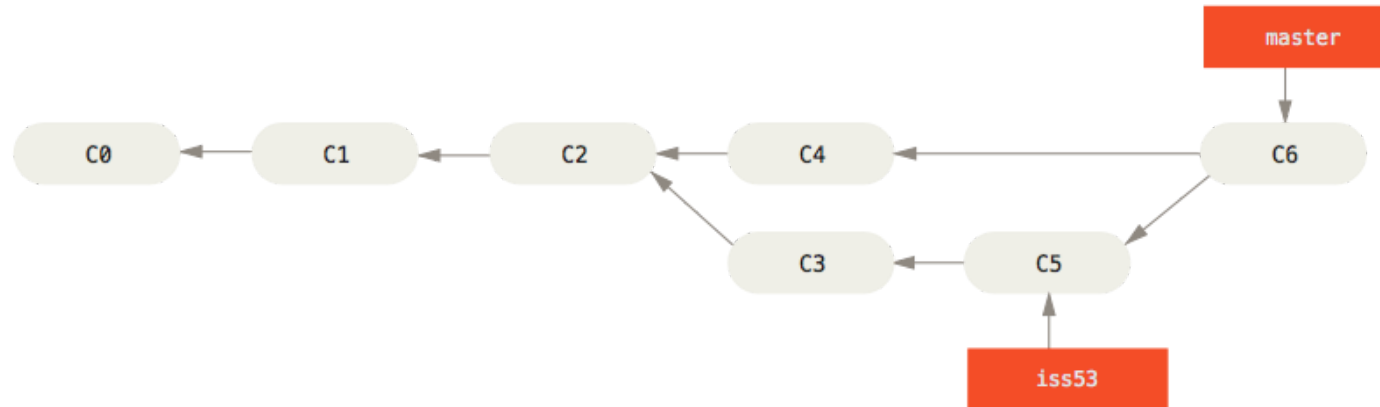
```
Merge branch 'iss53'  
  
# Por favor ingrese el mensaje del commit para sus cambios. Las líneas que comiencen  
# con '#' serán ignoradas, y un mensaje vacío aborta el commit.  
#  
# Fecha: Thu Jul 4 03:15:50 2019 -0300  
#  
# En la rama master  
# Cambios a ser confirmados:  
#   modificado: README.md  
#  
~  
~  
~  
~
```

merge

```
$ git checkout master
Cambiado a rama 'master'
$ git merge iss53
Merge made by the 'recursive' strategy.
 README.md | 4 ++++
 1 file changed, 4 insertions(+)
$
```

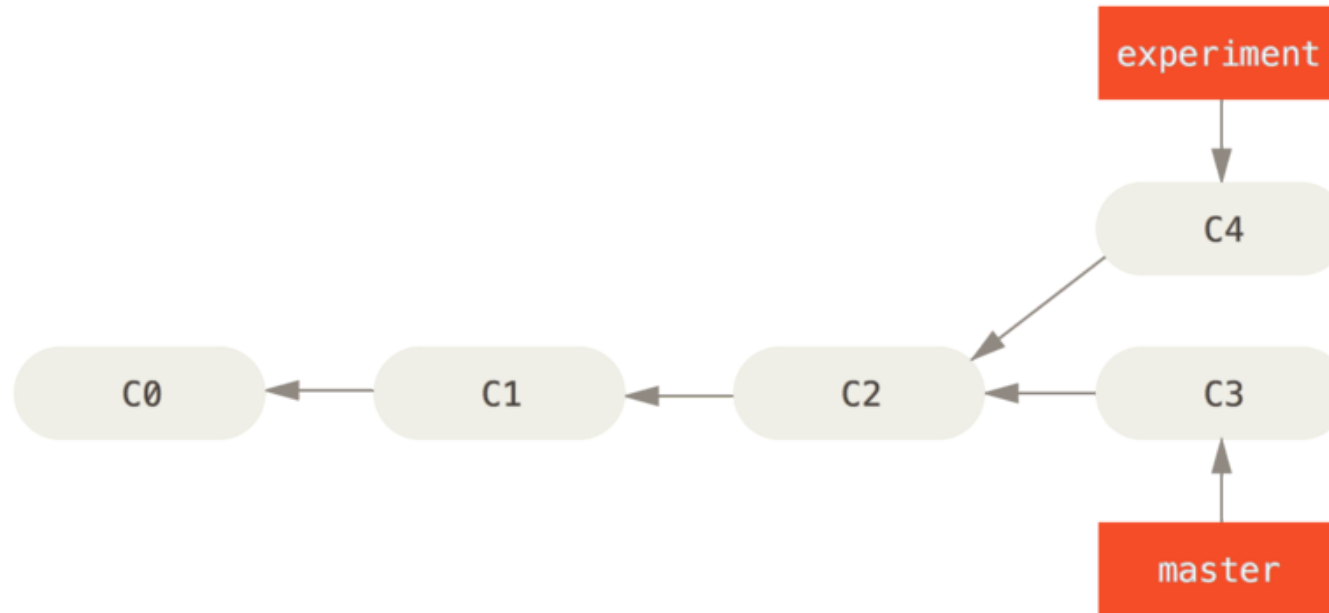

merge

merge



rebase

rebase



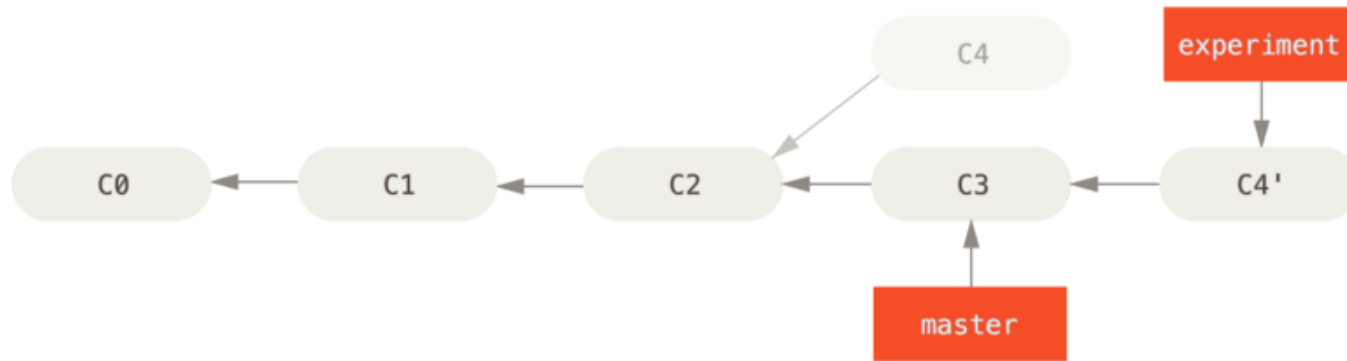
rebase

rebase

```
$ git checkout experiment  
$ git rebase master  
First, rewinding head to replay your work on top of it...  
Applying: added staged command
```

rebase

rebase



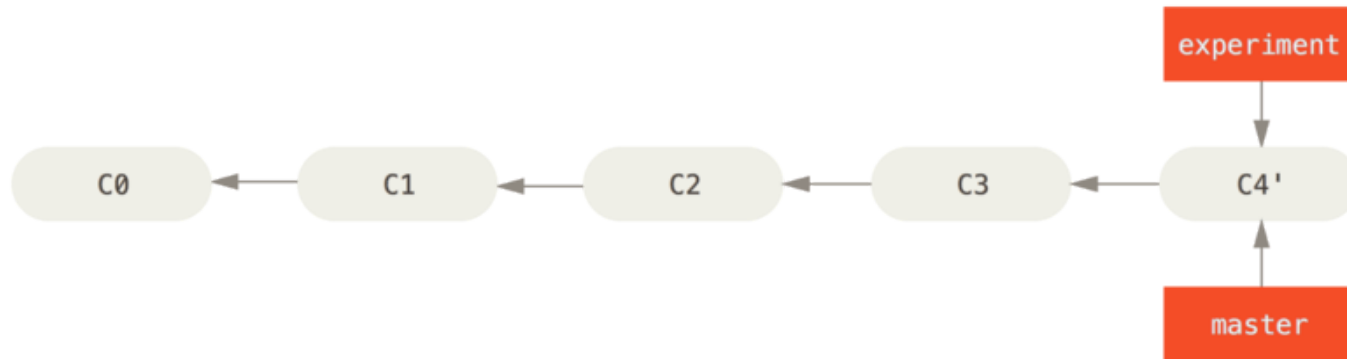
rebase

rebase

```
$ git checkout master  
$ git merge experiment
```

rebase

rebase



Bibliografía hasta acá

<https://git-scm.com/book/es/v2>



Gitflow

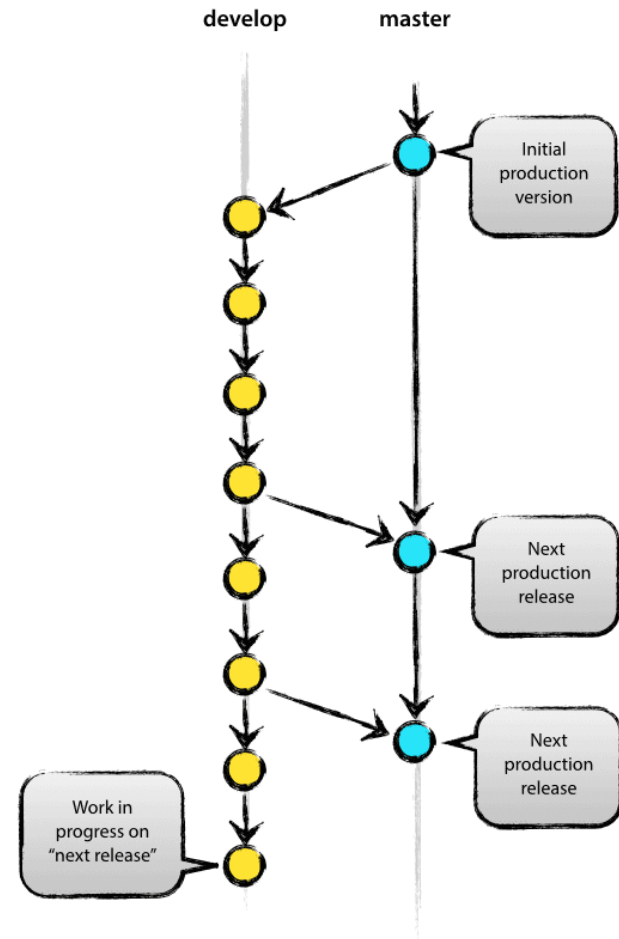
diseñado por Vincent Driessen (2010)

Flujo de trabajo basado en git que brinda un mayor control y organización en el proceso de integración continua.

Ideal para trabajar con desarrolladores junior.

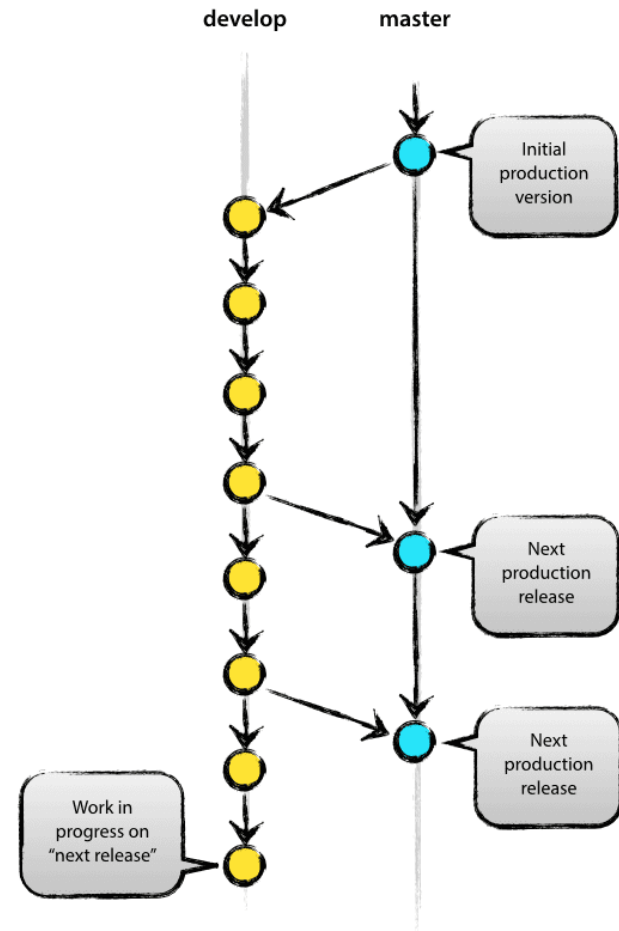
Ramas eternas

Ramas eternas

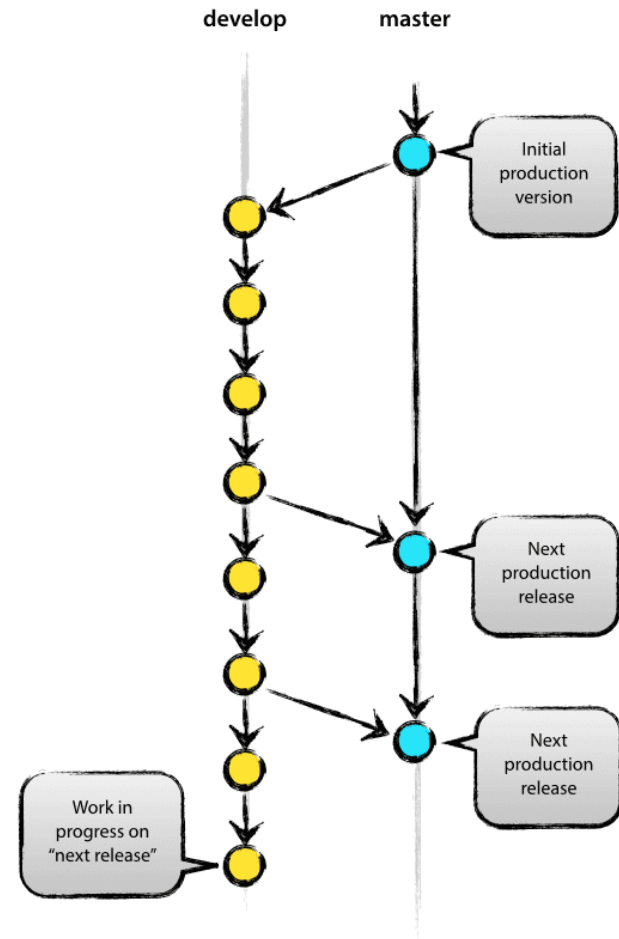


Ramas eternas

master solo contiene código listo para producción (uso en aplicación)

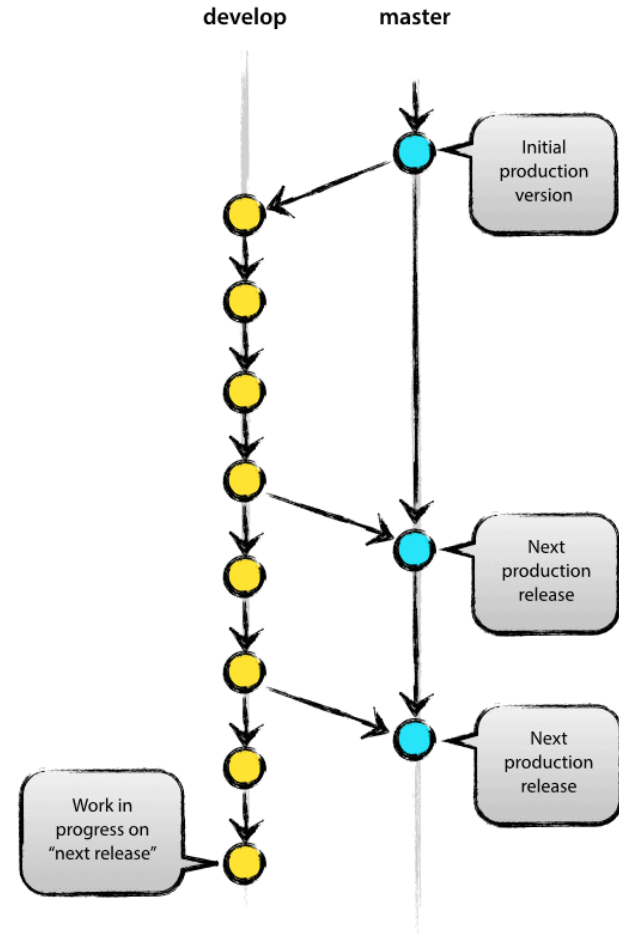


Ramas eternas

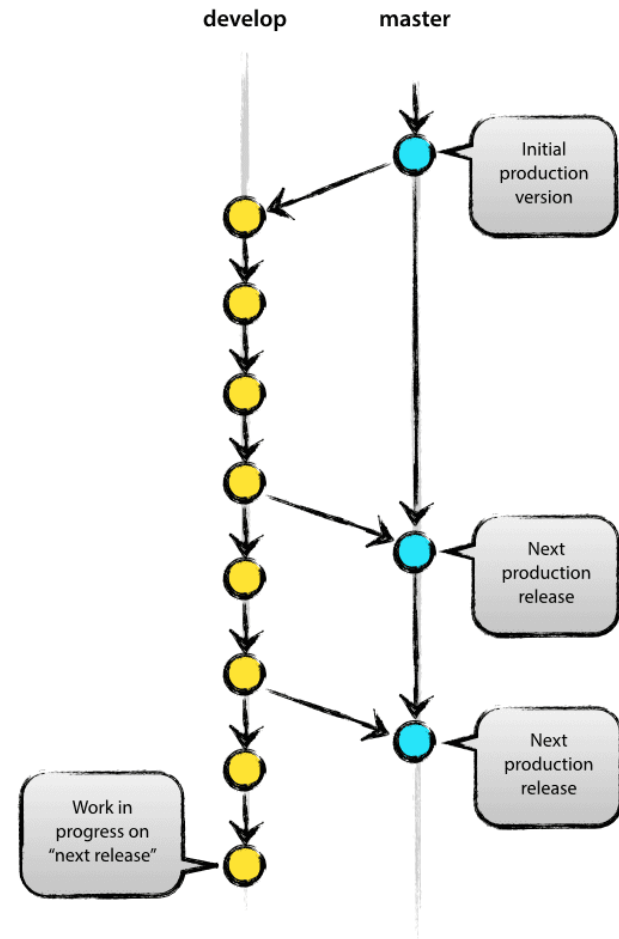


Ramas eternas

develop tiene código con los nuevos desarrollos y características que se incorporarán en el próximo release

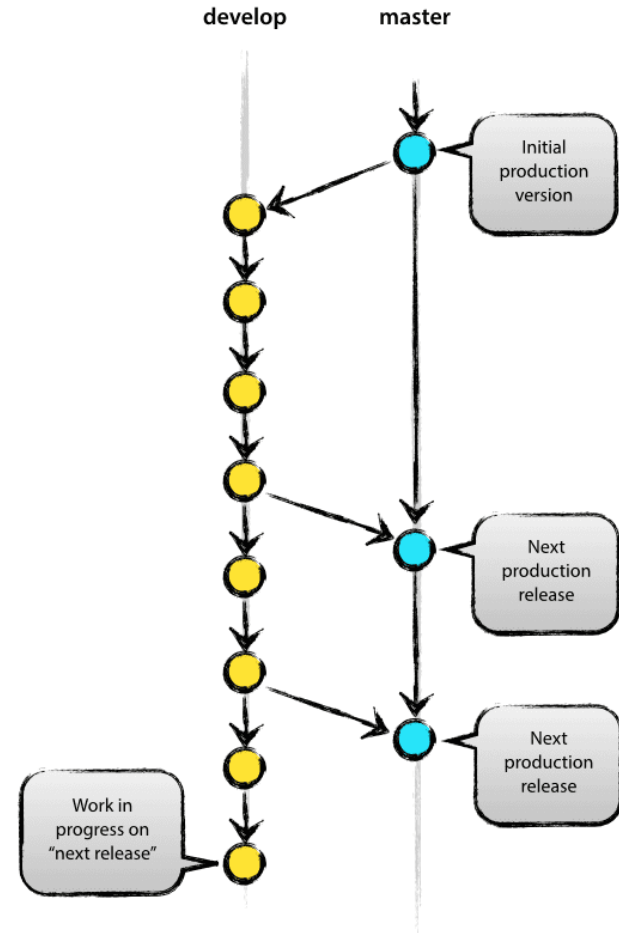


Ramas eternas



Ramas eternas

Solo cuando **develop** fue *testeada* lo suficiente es fusionado con **master**



Ramas de apoyo

Ramas de apoyo

Las ramas **feature** se usan para desarrollar nuevas características del programa

Ramas de apoyo

Las ramas **feature** se usan para desarrollar nuevas características del programa

La rama **release** es un paso intermedio entre **develop** y **master**. Generalmente aquí aparecen los números/nombres de versiones.

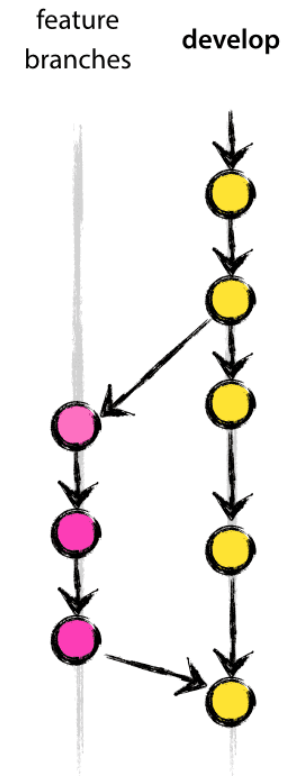
Ramas de apoyo

Las ramas **feature** se usan para desarrollar nuevas características del programa

La rama **release** es un paso intermedio entre **develop** y **master**. Generalmente aquí aparecen los números/nombres de versiones.

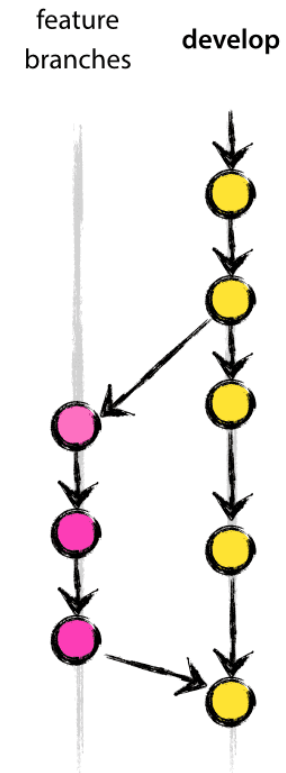
Las ramas **hotfix** resuelven bugs generalmente aparecidos en **master**.

Ramas de apoyo

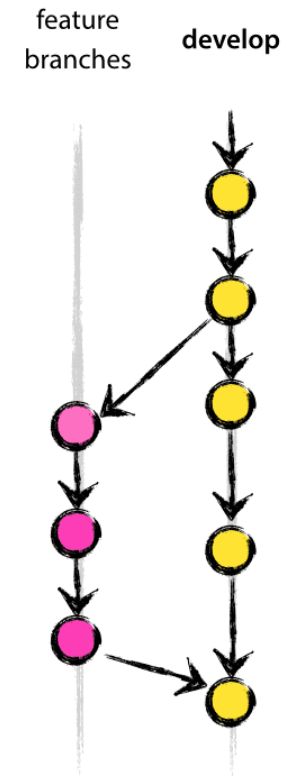


Ramas de apoyo

Las ramas **feature** siempre se desprenden y se fusionan en **develop**.

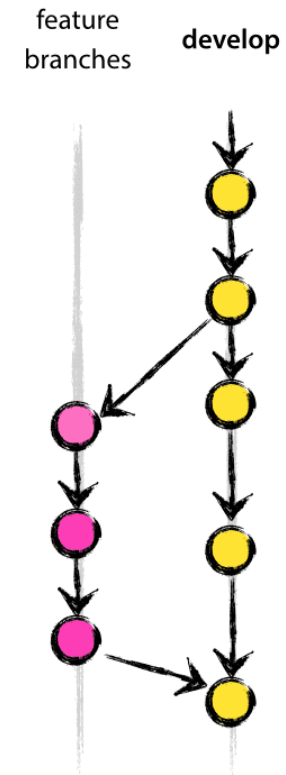


Ramas de apoyo



Ramas de apoyo

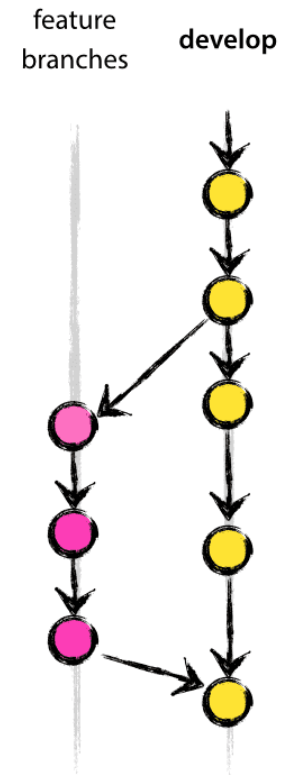
Para iniciar la rama...



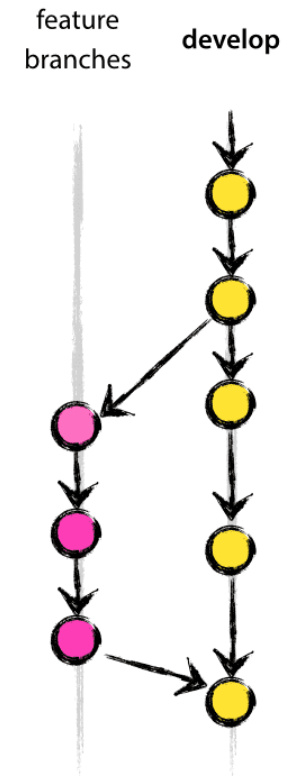
Ramas de apoyo

Para iniciar la rama...

```
$ git checkout -b myfeature develop  
Cambiado a nueva rama 'myfeature'  
$
```



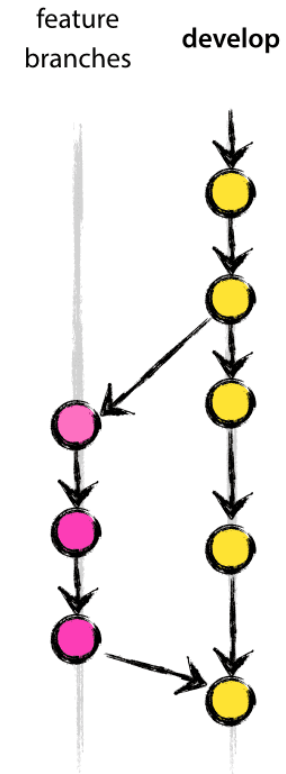
Ramas de apoyo



Ramas de apoyo

Para cerrarla...

```
$ git checkout develop
Cambiado a rama 'develop'
$ git merge --no-ff myfeature
Updating ea1b82a..05e9557
...
$ git branch -d myfeature
Eliminada la rama myfeature (era 05e9557).
...
```

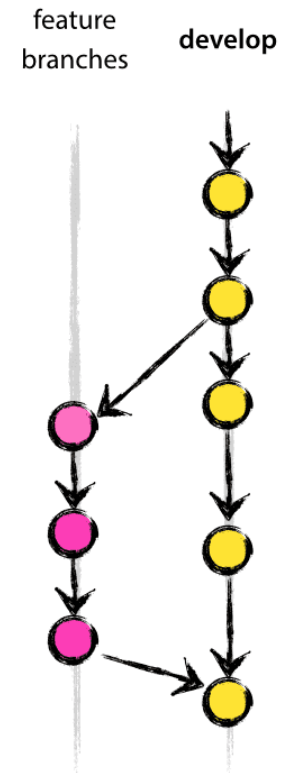


Ramas de apoyo

Para cerrarla...

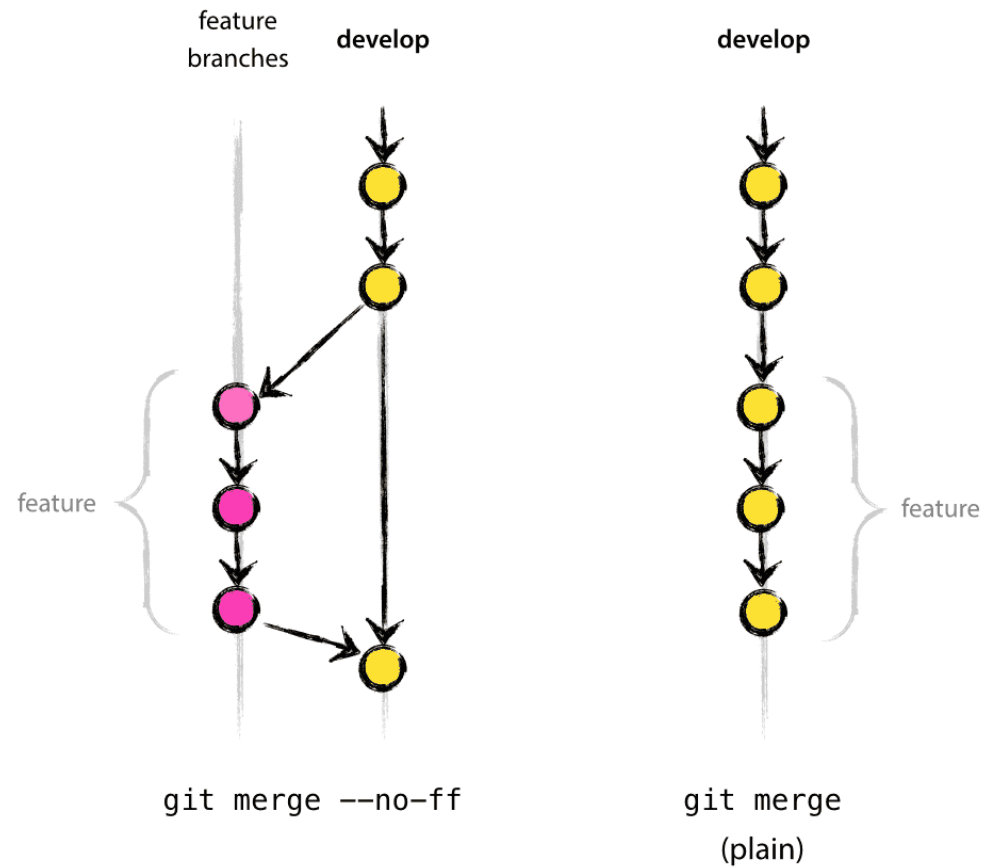
```
$ git checkout develop
Cambiado a rama 'develop'
$ git merge --no-ff myfeature
Updating ea1b82a..05e9557
...
$ git branch -d myfeature
Eliminada la rama myfeature (era 05e9557).
...
```

--no-ff obliga el commit aunque el merge sea por fast-forward



Ramas de apoyo

Ramas de apoyo



Ramas de apoyo

Ramas de apoyo

La rama **release** se desprende de **develop** y se fusiona en **master** y **develop**.

Ramas de apoyo

La rama **release** se desprende de **develop** y se fusiona en **master** y **develop**.

Para iniciarla...

Ramas de apoyo

La rama **release** se desprende de **develop** y se fusiona en **master** y **develop**.

Para iniciarla...

```
$ git checkout -b release-1.0 develop
Cambiado a nueva rama 'release-1.0'
$
```

Ramas de apoyo

La rama **release** se desprende de **develop** y se fusiona en **master** y **develop**.

Para iniciarla...

```
$ git checkout -b release-1.0 develop  
Cambiado a nueva rama 'release-1.0'  
$
```

Después de revisar todo se cierra

Ramas de apoyo

La rama **release** se desprende de **develop** y se fusiona en **master** y **develop**.

Para iniciarla...

```
$ git checkout -b release-1.0 develop
Cambiado a nueva rama 'release-1.0'
$
```

Después de revisar todo se cierra

```
$ git commit -a -m "Actualizada la versión a 1.0"
[release-1.0 74d9424] Actualizada la versión a 1.0
1 files changed, 1 insertions(+), 1 deletions(-)
```

Ramas de apoyo

Ramas de apoyo

```
$ git checkout master
Cambiado a rama 'master'
$ git merge --no-ff release-1.0
Merge made by the 'recursive' strategy.
...
$ git tag -a 1.0
$
```

Ramas de apoyo

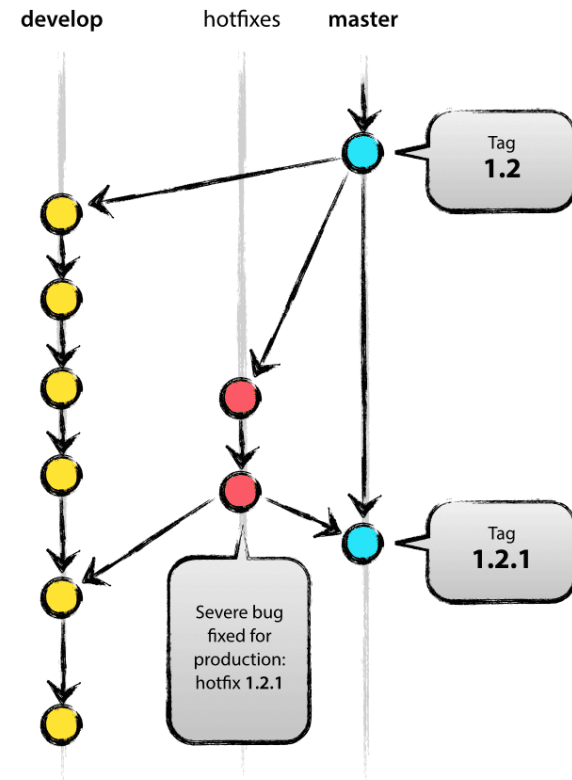
```
$ git checkout master
Cambiado a rama 'master'
$ git merge --no-ff release-1.0
Merge made by the 'recursive' strategy.
...
$ git tag -a 1.0
$ git checkout develop
Cambiado a rama 'develop'
$ git merge --no-ff release-1.0
Merge made by the 'recursive' strategy.
...
$
```

Ramas de apoyo

```
$ git checkout master
Cambiado a rama 'master'
$ git merge --no-ff release-1.0
Merge made by the 'recursive' strategy.
...
$ git tag -a 1.0
$ git checkout develop
Cambiado a rama 'develop'
$ git merge --no-ff release-1.0
Merge made by the 'recursive' strategy.
...
$ git branch -d release-1.0
Eliminada la rama release-1.0
$
```

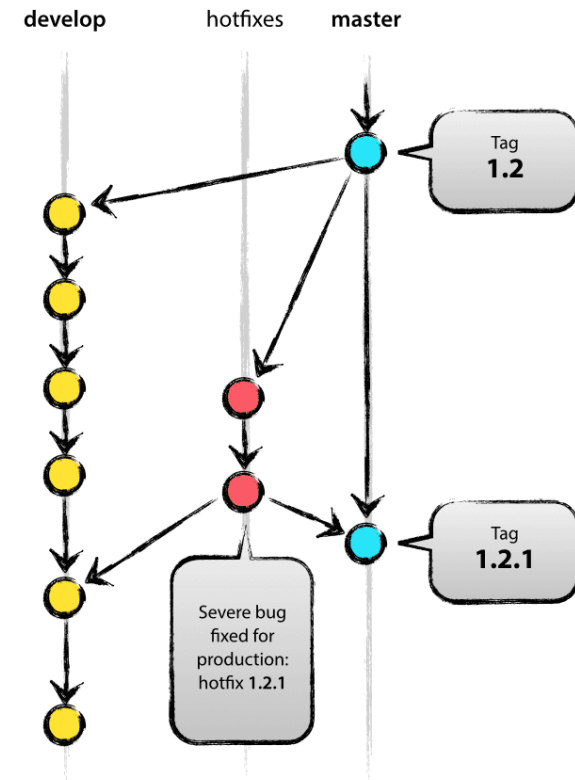
Ramas de apoyo

Ramas de apoyo

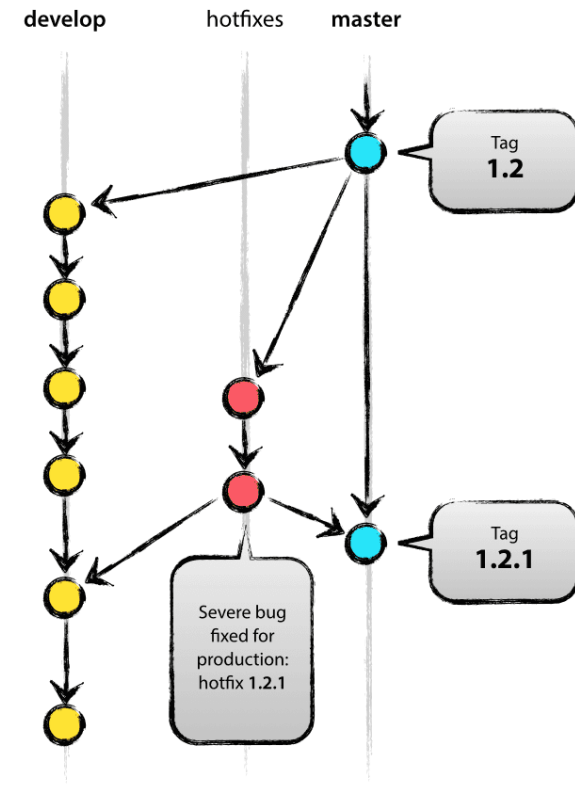


Ramas de apoyo

Las ramas **hotfix** se desprenden de **master** y se deben fusionar con **develop** y **master**

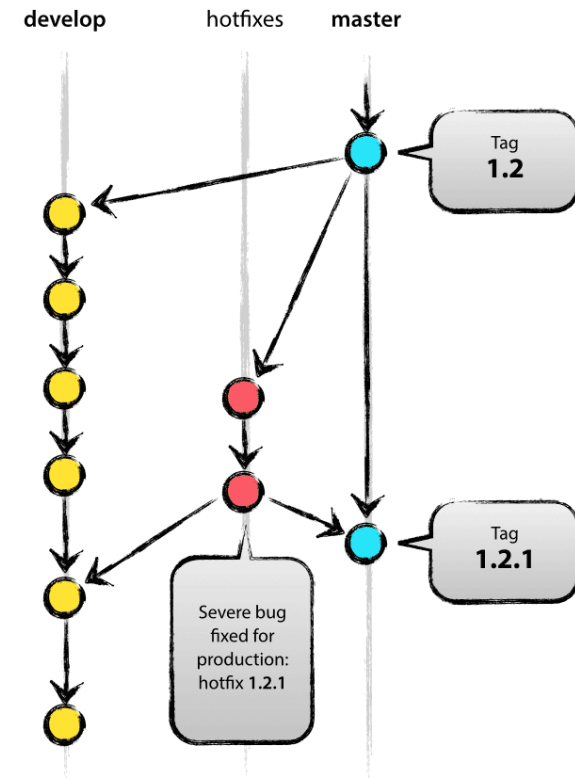


Ramas de apoyo



Ramas de apoyo

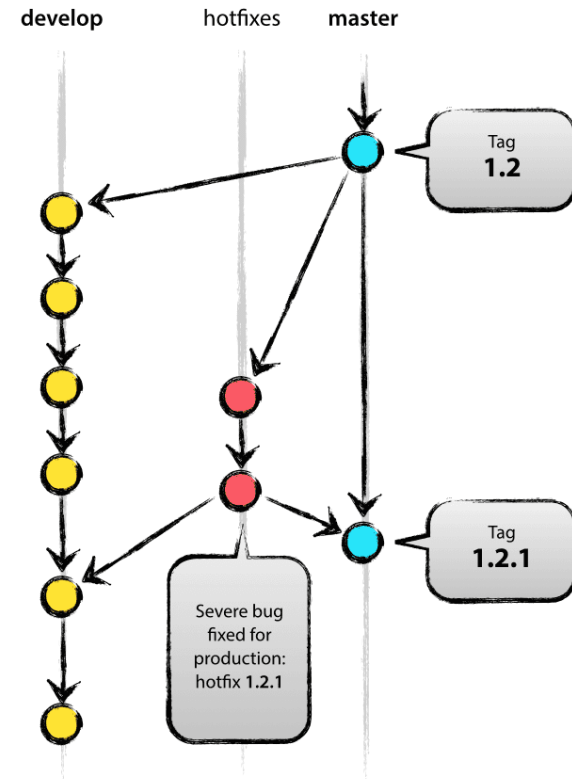
Para iniciar la rama...



Ramas de apoyo

Para iniciar la rama...

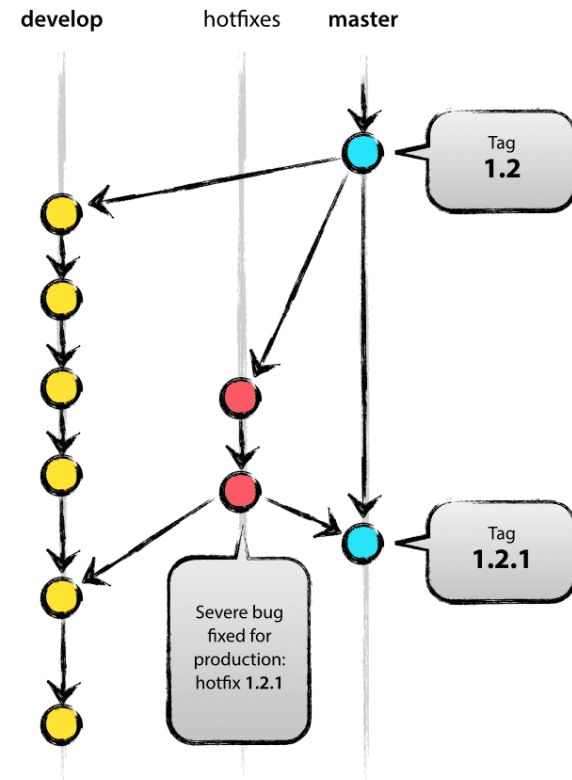
```
$ git checkout -b hotfix-1.2.1 master
Cambiado a rama 'hotfix-1.2.1'
...
```



Ramas de apoyo

Para iniciar la rama...

```
$ git checkout -b hotfix-1.2.1 master
Cambiado a rama 'hotfix-1.2.1'
...
$ git commit -a -m "Act. versión a 1.2.1"
...
```

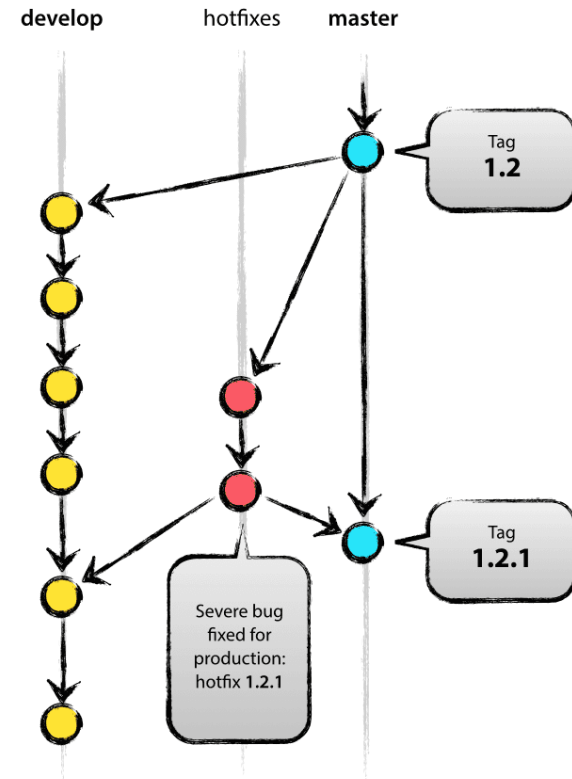


Ramas de apoyo

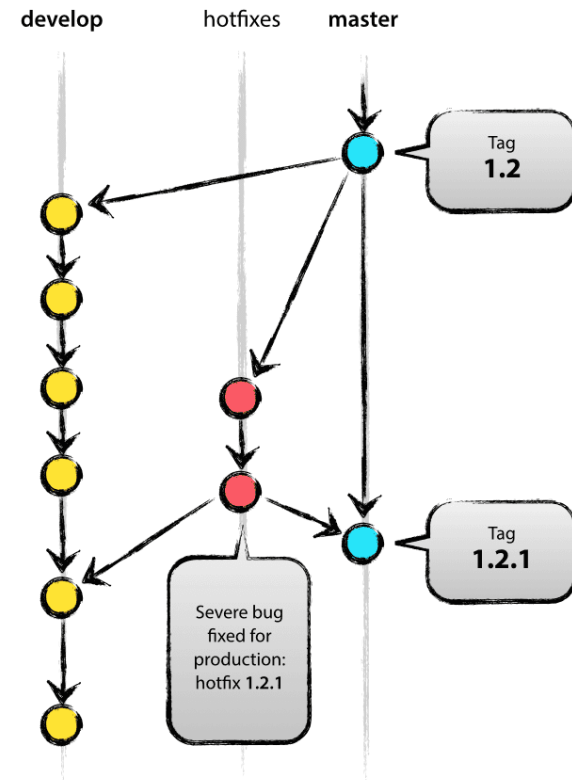
Para iniciar la rama...

```
$ git checkout -b hotfix-1.2.1 master
Cambiado a rama 'hotfix-1.2.1'
...
$ git commit -a -m "Act. versión a 1.2.1"
...
```

Luego de la corrección...

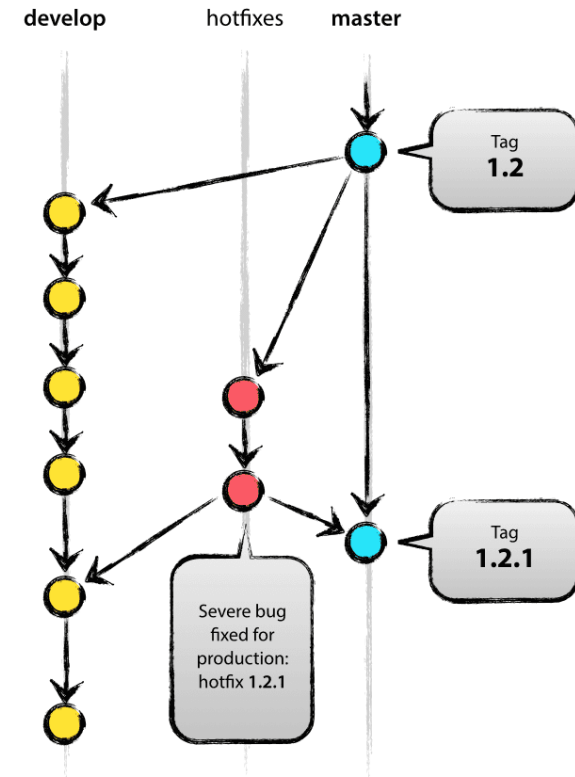


Ramas de apoyo



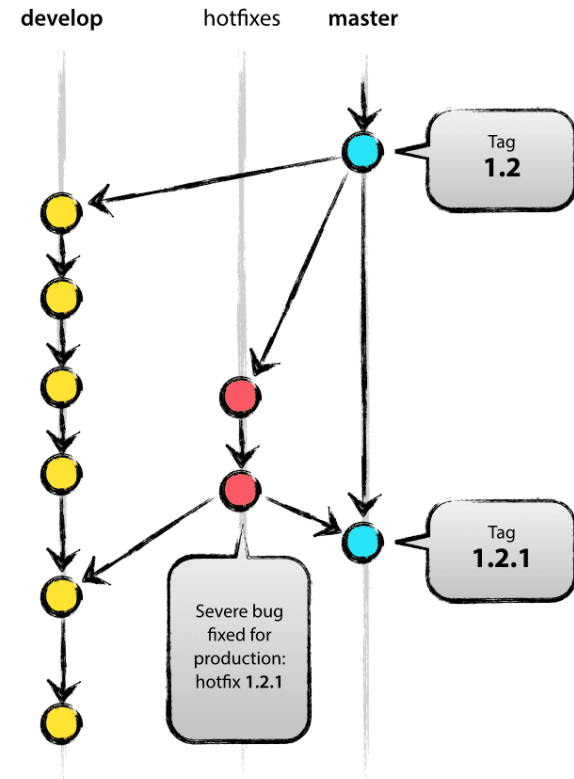
Ramas de apoyo

```
$ git commit -m "Soluciona el bug"  
[hotfix-1.2.1 abbe5d6] Soluciona el bug  
5 files changed, 32 insertions(+)  
$
```



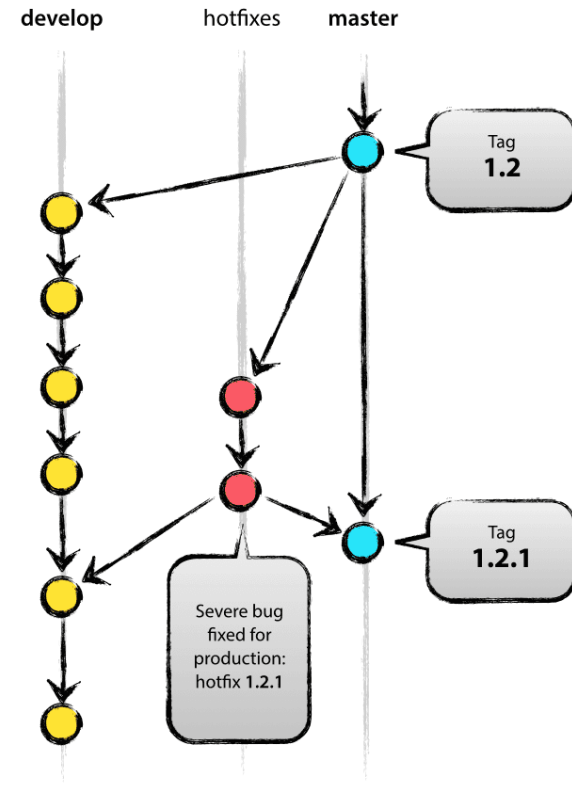
Ramas de apoyo

```
$ git commit -m "Soluciona el bug"
[hotfix-1.2.1 abbe5d6] Soluciona el bug
5 files changed, 32 insertions(+)
$ git checkout master
Cambiado a rama 'master'
$ git merge --no-ff hotfix-1.2.1
Merge made by recursive.
(Summary of changes)
$ git tag -a 1.2.1
$
```



Ramas de apoyo

```
$ git commit -m "Soluciona el bug"
[hotfix-1.2.1 abbe5d6] Soluciona el bug
5 files changed, 32 insertions(+)
$ git checkout master
Cambiado a rama 'master'
$ git merge --no-ff hotfix-1.2.1
Merge made by recursive.
(Summary of changes)
$ git tag -a 1.2.1
$ git checkout develop
Cambiado a rama 'develop'
$ git merge --no-ff hotfix-1.2.1
Merge made by recursive.
(Summary of changes)
$ git branch -d hotfix-1.2.1
Eliminada la rama hotfix-1.2.1
```



Importante

Importante

Solo un encargado de hacer los merge hacia **master**.

Importante

Solo un encargado de hacer los merge hacia **master**.

Trabajar con *pull-request* o *merge-request*.

Importante

Solo un encargado de hacer los merge hacia **master**.

Trabajar con *pull-request* o *merge-request*.

Es preferible que las ramas **features** también los usen.

Importante

Solo un encargado de hacer los merge hacia **master**.

Trabajar con *pull-request* o *merge-request*.

Es preferible que las ramas **features** también los usen.

Pueden haber más autorizados a aceptarlos.

Lo que quedó en el tintero

Lo que quedó en el tintero

Más comandos de git (grep, stash, bisect)

Lo que quedó en el tintero

Más comandos de git (grep, stash, bisect)

Jugar con las versiones remotas

Lo que quedó en el tintero

Más comandos de git (grep, stash, bisect)

Jugar con las versiones remotas

Ver las extensiones `git-flow`

¡Gracias!

¿Preguntas?

claudiojpaz@gmail.com