

Introducción a Git

Carlos Ruiz Pastor



Objetivo

Entender el funcionamiento de Git y aprender los comandos básicos para usar Git en las prácticas.

¿Qué es Git?

Es un **sistema de control de versiones distribuido**, que nos permite gestionar los cambios que hacemos sobre un directorio.



Open Source



Velocidad



Escalable



Robusto



Seguro



Económico

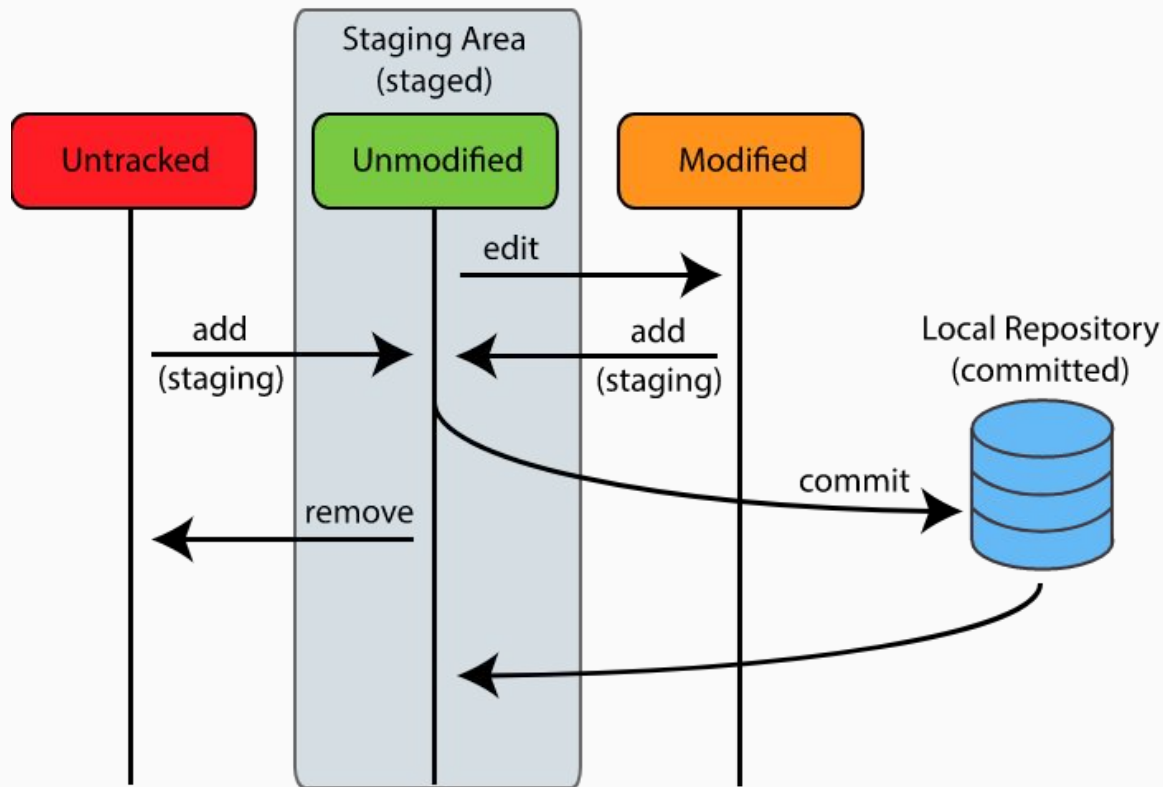
¿Cómo funciona?

Git almacena instantáneas de un mini sistema de archivos.

Modified: Cambios realizados, pero que aún no se han confirmado en la base de datos local.

Staged/Unmodified: Preparado, el momento en que marcas un archivo modificado para que sea confirmado y almacenado en tu base de datos local.

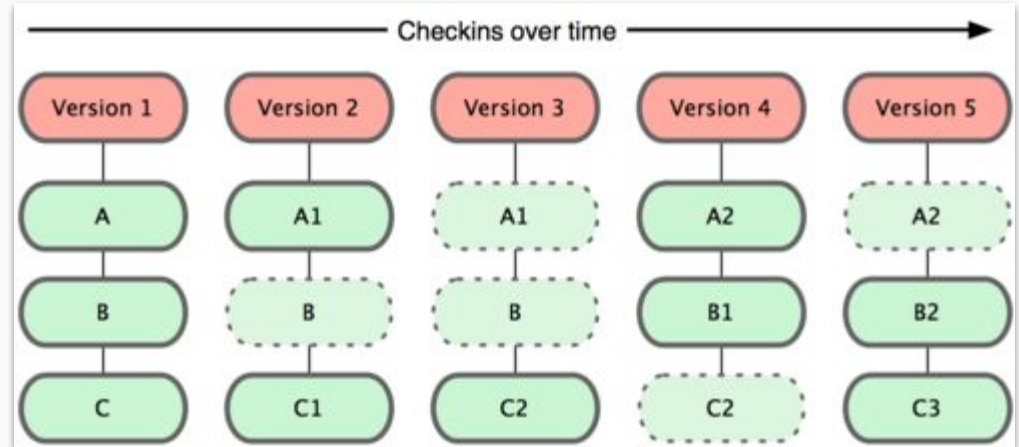
Committed: Cambios confirmados, en este momento tus cambios ya están almacenados en tu base de datos local.



¿Cómo funciona?

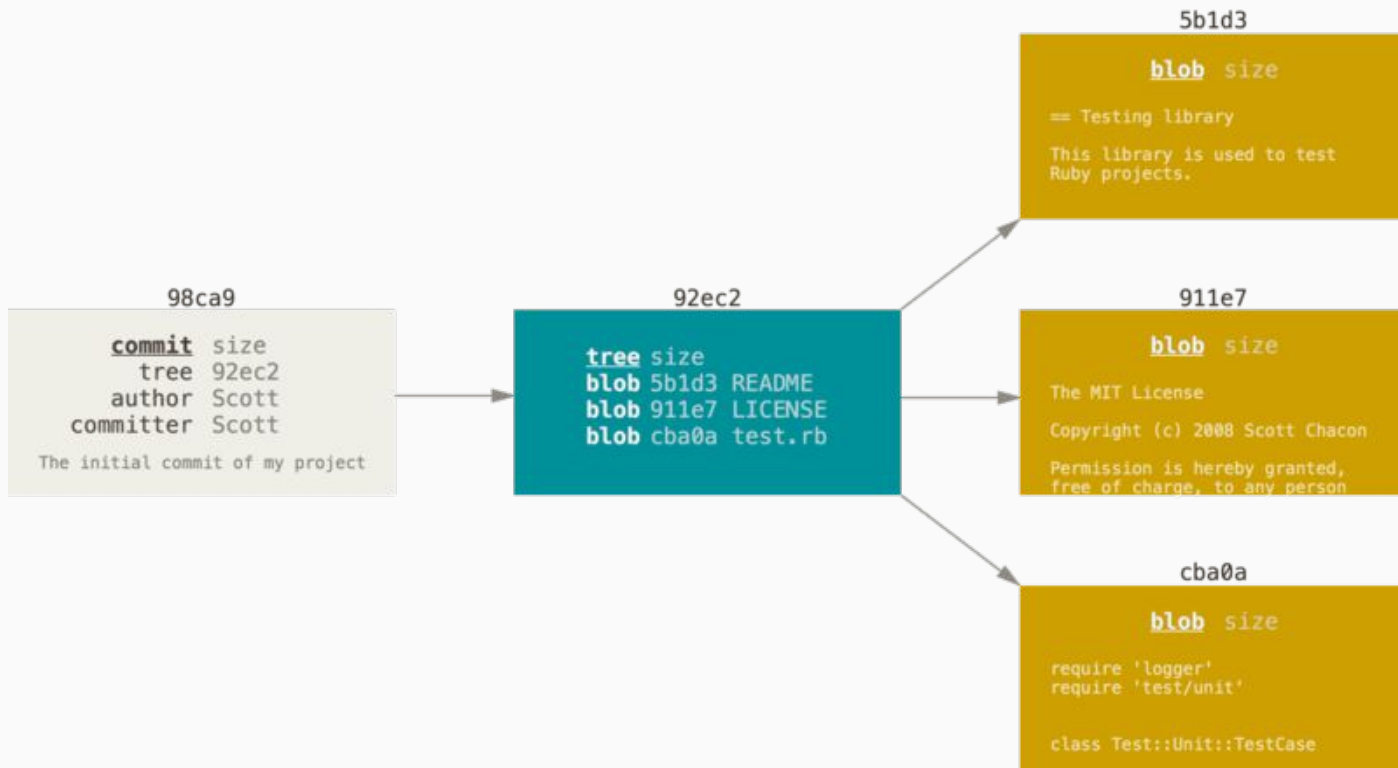
Git almacena instantáneas de un mini sistema de archivos.

En cada **commit** toma una foto al aspecto de tu proyecto en ese momento y crea una referencia a esa instantánea, si un archivo no cambió git no almacena el nuevo archivo sino que crea un enlace a la imagen anterior.



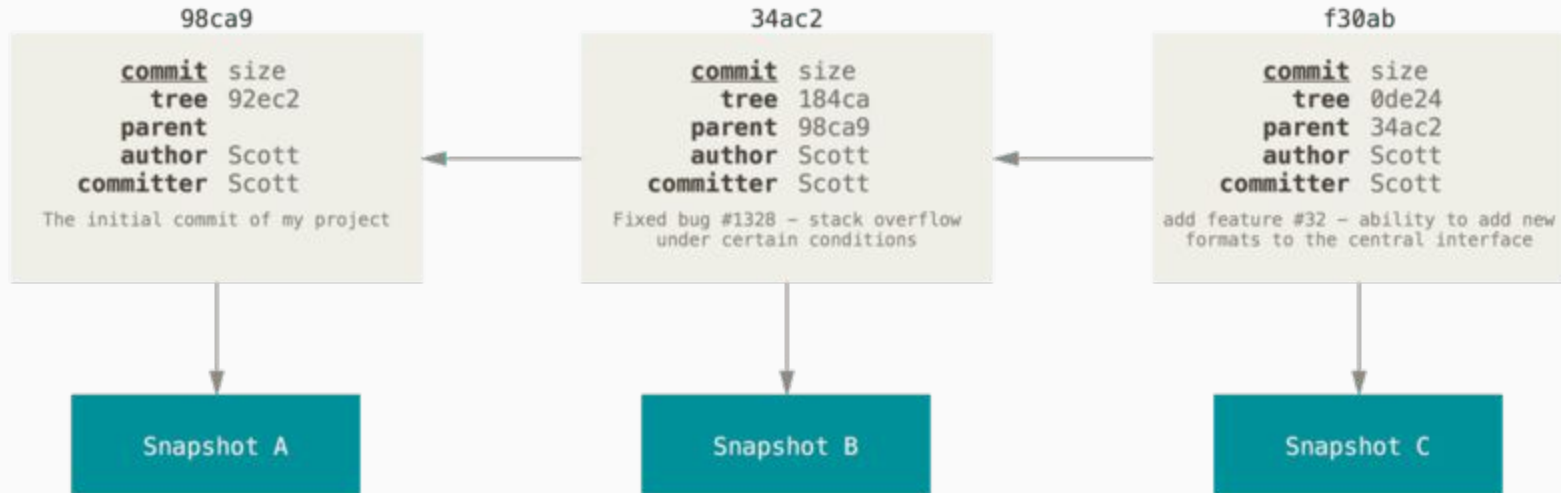
```
$ git add README test.rb LICENSE
```

```
$ git commit -m 'The initial commit of my project'
```



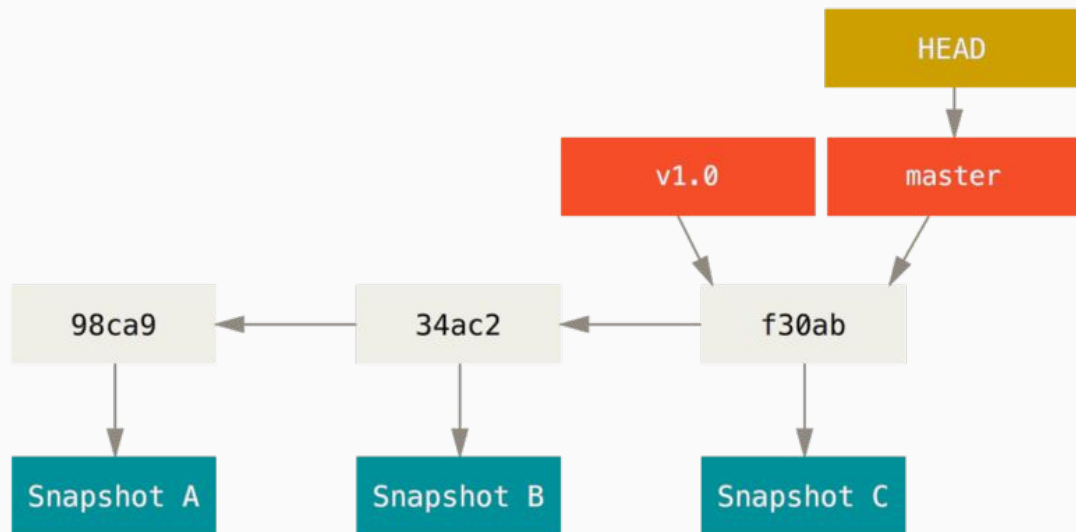
```
$ git commit -m 'Fixed bug #1328 - stack overflow under certain conditions'
```

```
$ git commit -m 'add feature #32 - ability to add new formats to the central interface'
```

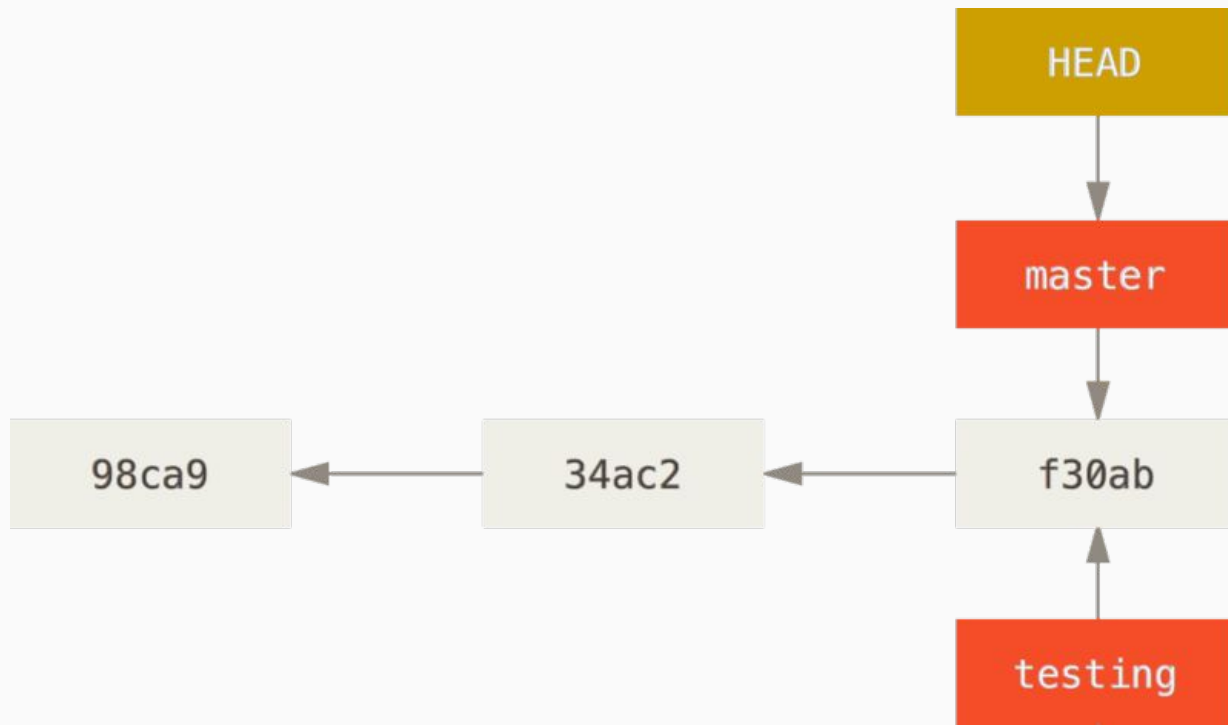


Ramas

- **Una rama es un puntero a estos commits.**
- **La rama por defecto** al hacer un commit **es la rama master** y apunta al último commit que hayas hecho.
- **HEAD** es un puntero que apunta al commit donde estamos trabajando. Puede apuntar a una rama o directamente a un commit.



\$ git branch testing

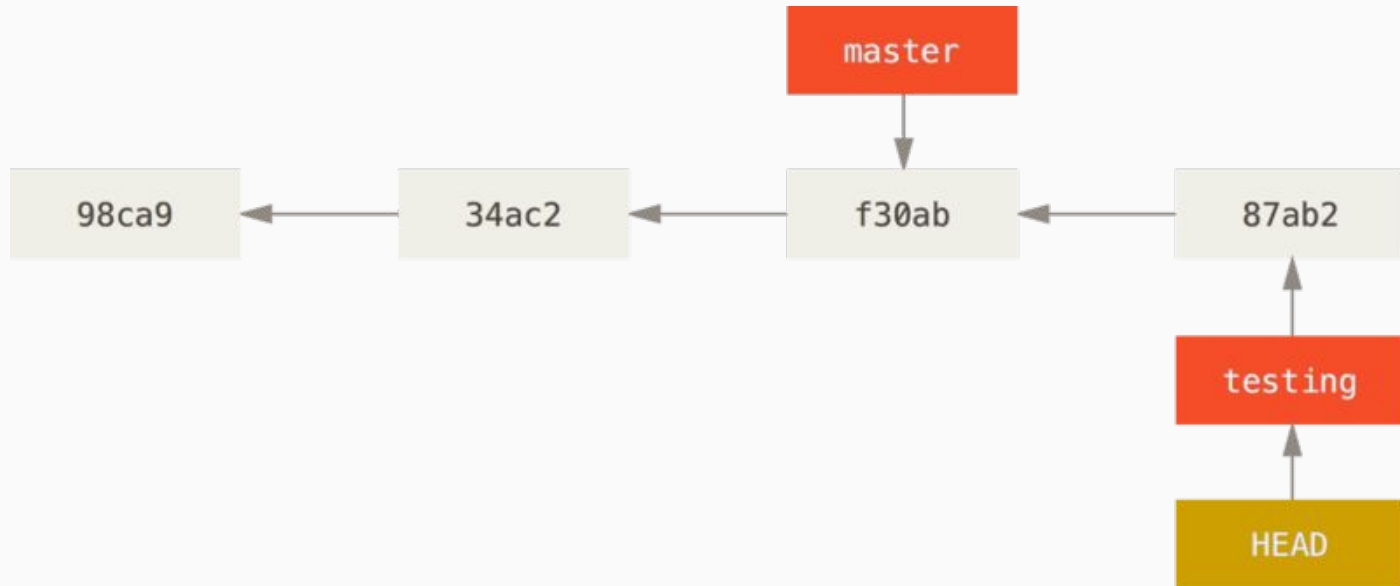


```
$ git checkout testing
```

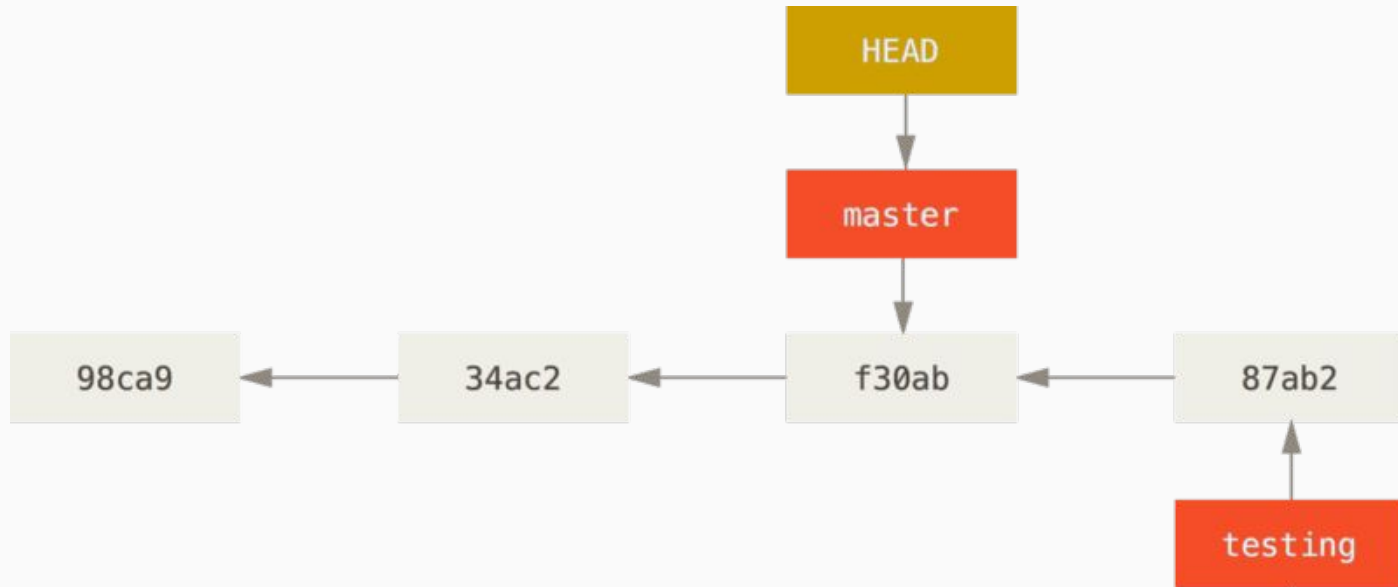


```
$ vim test.rb
```

```
$ git commit -a -m 'made a change'
```

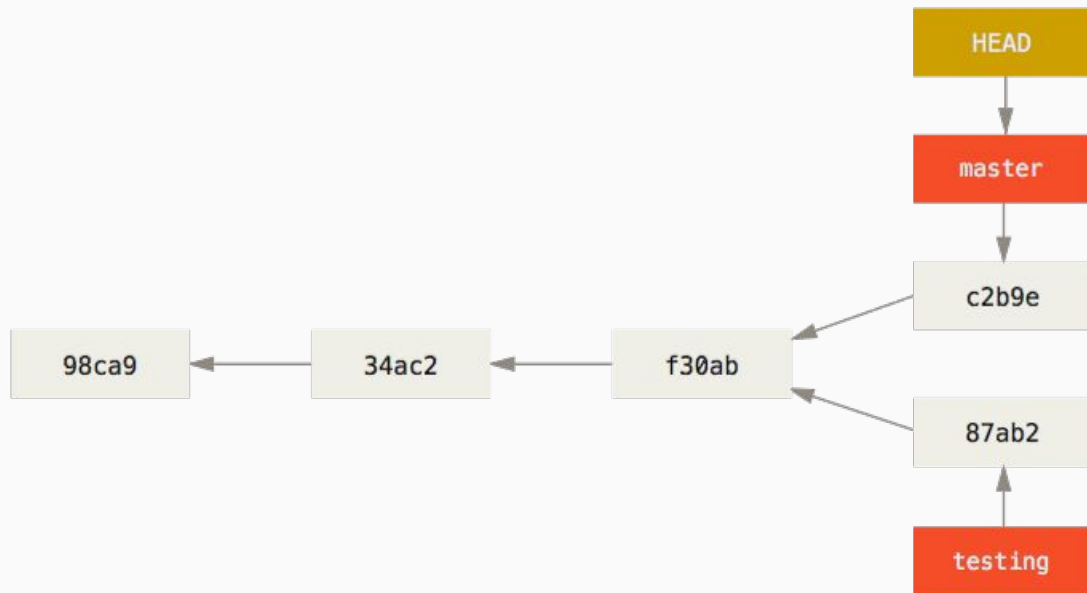


```
$ git checkout master
```



```
$ vim test.rb
```

```
$ git commit -a -m 'made other changes'
```

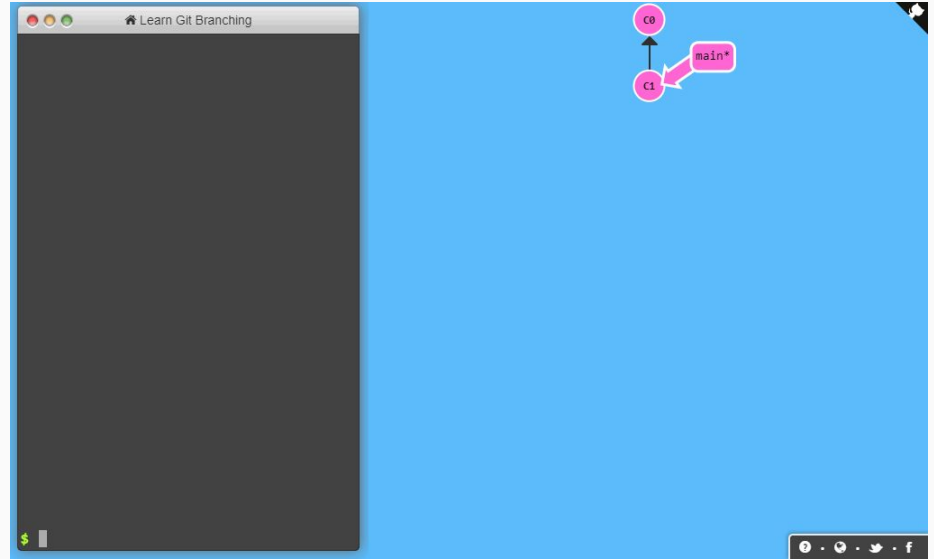


Comandos básicos

- git add
- git commit
- git branch
- git checkout
- git merge
- git rebase
- git push
- git pull
- git fetch

Ejercicios: Learn Git Branching

<https://learngitbranching.js.org/>





⚙ Selección de nivel

Principal

Remota

Secuencia introductoria

Una breve introducción a la mayoría de los comandos de git

1: *Introducción a los commits de Git*

1

2

3

4

Acelerando

La próxima ración de git. Espero que estés hambriento

1: *Desatachea tu HEAD*

1

2

3

4



⚙ Selección de nivel

Principal

Remota

Push y Pull -- Git Remotes!

Hora de compartir vuestros 1's y 0's, chicos; programar se volvió social!

1: *Introducción a clone*

1

2

3

4

5

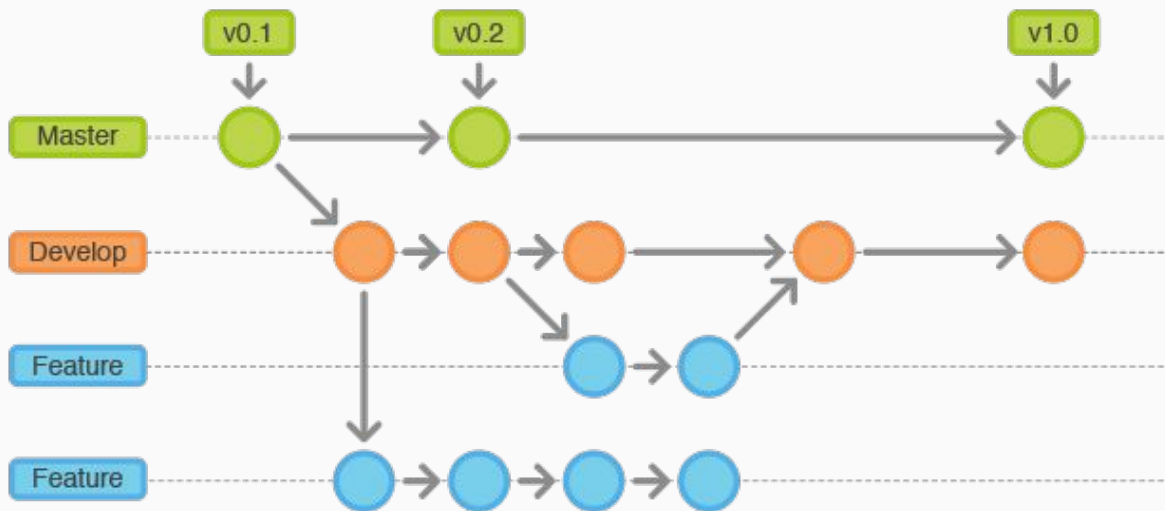
6

7

Ramas Recomendadas

“Crea ramas al principio y hazlo también de manera frecuente”

[Gitflow Workflow](#)



TAG

Os voy a pedir que pongáis una tag anotada con el nombre “entrega” en el commit de la entrega

¿Dudas?