



Contenido

- Conceptos Generales
- Instalación
- Repositorios
- Status
- Commit
- Branch
- Merge
- GitFlow
- Pull Request



Conceptos Generales



Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones.

Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

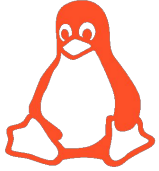
Web: <https://git-scm.com/>

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

Guia rapida : <https://rogerdudler.github.io/git-guide/index.es.html>



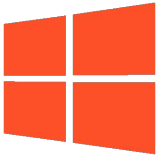
Instalación



Git for Linux



Git for Mac



Git for Windows



Repositorios

Un repositorio es un proyecto de git. Puedes obtener un proyecto Git de dos maneras. La primera es tomar un proyecto o directorio existente e importarlo en Git. La segunda es clonar un repositorio existente en Git desde otro servidor.

Iniciar repositorio: `$ git init`

Clonar un repositorio: `$ git clone https://github.com/libgit2/libgit2`

Servicios de repos: <http://github.com>, <http://gitlab.com>, <http://bitbucket.org>



git status

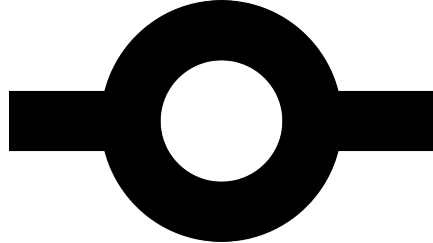
git status permite verificar el estado de nuestro repositorio.

```
$ git status
{ On branch master } Branch Actual
{ nothing to commit, working directory clean } Repositorio limpio
```

```
$ git status
On branch master
{ Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
  new file:   README
} Archivos con seguimiento en el repositorio
{ Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
  modified:   CONTRIBUTING.md
} Archivos sin seguimiento en el repositorio
```



Commit



El comando `git commit` captura una instantánea de los cambios preparados en ese momento del proyecto. Las instantáneas confirmadas pueden considerarse como versiones "seguras" de un proyecto: Git no las cambiará nunca a no ser que se lo pidas expresamente.

Agrega un archivo

→

```
$ git add <archivo_pendiente>
```

Verifica archivos a commitear

→

```
$ git status  
On branch master  
Changes to be committed:  
  (use "git reset HEAD <file>..." to unstage)
```

```
renamed:    README.md -> README
```

Crea commit

→

```
$ git commit -m "<mensaje asociado a commitear>"
```

Quita archivos del commitear

→ a

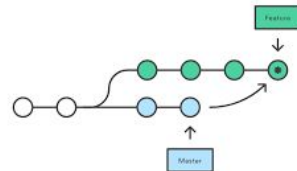
```
$ git checkout -- CONTRIBUTING.md
```



Crear, Listar, Borrar y editar ramas:	\$ git branch
Crear una rama desde la actual:	\$ git checkout -b <nombre_nuevo_branch>
Cambiarse a otra rama:	\$ git checkout <branch_al_que_queremos_pasar>



Merge



El merge combinará varias secuencias de confirmaciones en un historial unificado. En los casos de uso más frecuentes, `git merge` se utiliza para combinar dos ramas.

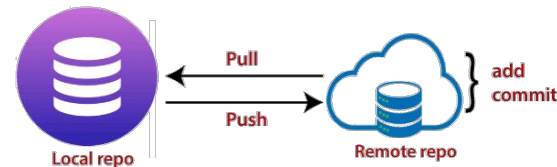
Combinar una rama `<rama_a>` en nuestra rama actual `<rama_b>`:

```
$ git merge rama_a
```

NOTA: Al realizar un merge pueden presentarse conflictos que tendremos que resolver para que se complete el merge.



Remote



Git tiene la posibilidad de sincronizarse con una copia del repositorio de forma remota.

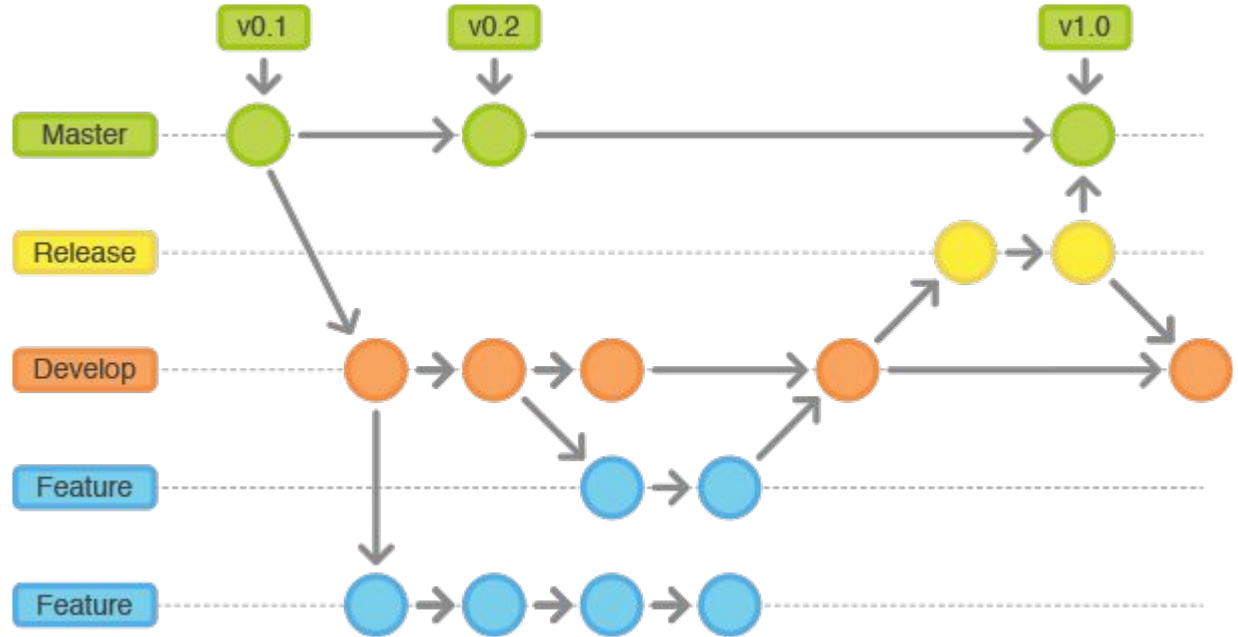
- Configurar repositorio remoto:
 - `$ git remote add origin <server>`
- Enviar cambios al repositorio remoto:
 - `$ git push origin <branch>`
- Traer cambios del repositorio remoto:
 - `$ git pull origin <branch>`

NOTA: El nombre origin es solo un identificador que usamos el cual puede cambiar o tener más de 1 configurado.



GitFlow

Gitflow es un diseño de flujo de trabajo de Git que fue publicado por primera vez y popularizado por Vincent Driessen en nvie. El flujo de trabajo de Gitflow define un modelo estricto de ramificación diseñado alrededor de la publicación del proyecto. Proporciona un marco sólido para gestionar proyectos más grandes.



Pull Request

Cuando un desarrollador termina de crear (y probar) ya sea una nueva funcionalidad o corrección de bug, solicita integrar su desarrollo al repositorio principal. Esta solicitud se le conoce como pull request (o PR) y pide al resto del equipo desarrollador revisar el código para que ellos:

- Proporcionen sus comentarios, retroalimentación o bien...
- Que puedan aprobar o rechazar el PR

Una vez aprobado el pull request, el desarrollador u otro integrante puede unir el nuevo desarrollo al desarrollo principal.



