

# Ramas

## Administración de ramas

### Crear una nueva rama

Cuando vamos a trabajar en una nueva funcionalidad, es conveniente hacerlo en una nueva rama, para no modificar la rama principal y dejarla inestable. Aunque la orden para manejar ramas es `git branch` podemos usar también `git checkout`.

Vamos a crear una nueva rama:

```
git branch hola
```

#### Info

Si usamos `git branch` sin ningún argumento, nos devolverá la lista de ramas disponibles.

La orden anterior no devuelve ningún resultado y tampoco nos cambia de rama, para eso debemos usar *checkout*:

```
$ git checkout hola
Switched to branch 'hola'
```

#### Tip

Hay una forma más rápida de hacer ambas acciones en un solo paso. Con el parámetro `-b` de `git checkout` podemos cambiarnos a una rama que, si no existe, se crea instantáneamente.

```
$ git checkout -b hola
Switched to a new branch 'hola'
```

## Modificaciones en la rama secundaria

Añadimos un nuevo archivo en el directorio `lib` llamado `HolaMundo.php`:

```
<?php

class HolaMundo
```

```

{
    private $nombre;

    function __construct($nombre)
    {
        $this->nombre = $nombre;
    }

    function __toString()
    {
        return sprintf ("Hola, %s.\n", $this->nombre);
    }
}
?>

```

Y modificamos `hola.php`:

```

<?php
// Autor: Sergio Gómez <sergio@uco.es>
// El nombre por defecto es Mundo
require('HolaMundo.php');

$nombre = isset($argv[1]) ? $argv[1] : "Mundo";
print new HolaMundo($nombre);
?>

```

Podríamos confirmar los cambios todos de golpe, pero lo haremos de uno en uno, con su comentario.

```

$ git add lib/HolaMundo.php

$ git commit -m "Añadida la clase HolaMundo"
[hola 6932156] Añadida la clase HolaMundo
1 file changed, 17 insertions(+)
create mode 100644 lib/HolaMundo.php

$ git add lib/hola.php

$ git commit -m "hola usa la clase HolaMundo"
[hola 9862f33] hola usa la clase HolaMundo
1 file changed, 3 insertions(+), 1 deletion(-)

```

Y ahora con la orden `git checkout` podemos movernos entre ramas:

```

$ git checkout master
Switched to branch 'master'

$ git checkout hola
Switched to branch 'hola'

```

## Modificaciones en la rama master

Podemos volver y añadir un nuevo archivo a la rama principal:

```
$ git checkout master
Switched to branch 'master'
```

Creamos un archivo llamado `README.md` en la raíz de nuestro proyecto con el siguiente contenido:

```
# Curso de GIT

Este proyecto contiene el curso de introducción a GIT
```

Y lo añadimos a nuestro repositorio en la rama en la que estamos:

```
$ git add README.md

$ git commit -m "Añadido README.md"
[master c3e65d0] Añadido README.md
1 file changed, 3 insertions(+)
create mode 100644 README.md
```

Vamos a usar nuestro comando `git hist` con el modificador `--all` que nos mostrará los cambios en todas las ramas.

```
$ git hist --all
* c3e65d0 2013-06-16 | Añadido README.md (HEAD -> master) [Sergio Gómez]
| * 9862f33 2013-06-16 | hola usa la clase HolaMundo (hola) [Sergio Gómez]
| * 6932156 2013-06-16 | Añadida la clase HolaMundo [Sergio Gómez]
| /
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio Gómez]
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto
(tag: v1) [Sergio Gómez]
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta)
[Sergio Gómez]
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]
```

Y vemos como `git hist --all` muestra la bifurcación en nuestro código.

## Fusión de ramas y resolución de conflictos

### Mezclar ramas

Podemos incorporar los cambios de una rama a otra con la orden `git merge`. Primero nos posicionamos en la rama en la que queremos incorporar los cambios. En nuestro caso "hola".

```
$ git checkout hola
```

```
Switched to branch 'hola'
```

Y en esta rama ejecutamos el `git merge nombrerama` con el nombre de la rama cuyos cambios queremos incorporar a la actual.

```
$ git merge master
Merge made by the 'ort' strategy.
 README.md | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 README.md

$ git hist --all
* 9c6ac06 2013-06-16 | Merge branch 'master' into hola (HEAD -> hola)
[Sergio Gómez]
|\
| * c3e65d0 2013-06-16 | Añadido README.md [Sergio Gómez]
* | 9862f33 2013-06-16 | hola usa la clase HolaMundo [Sergio Gómez]
* | 6932156 2013-06-16 | Añadida la clase HolaMundo [Sergio Gómez]
|/
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio Gómez]
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto
(tag: v1) [Sergio Gómez]
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta)
[Sergio Gómez]
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]
```

De esa forma se puede trabajar en una rama secundaria incorporando los cambios de la rama principal o de otra rama.

## Resolver conflictos

Un conflicto es cuando se produce una fusión que Git no es capaz de resolver. Vamos a modificar la rama master para crear uno con la rama hola.

```
$ git checkout master
Switched to branch 'master'
```

Modificamos nuestro archivo *hola.php* de nuevo:

```
<?php
// Autor: Sergio Gómez <sergio@uco.es>
print "Introduce tu nombre:";
$nombre = trim(fgets(STDIN));
@print "Hola, {$nombre}\n";
?>
```

Y guardamos los cambios:

```
$ git add lib/hola.php

$ git commit -m "Programa interactivo"
[master 9c85275] Programa interactivo
 1 file changed, 2 insertions(+), 2 deletions(-)

$ git hist --all
```

## Comprueba el resultado

```
$ git hist --all
* 8ca56f7 2013-06-16 | Programa interactivo (HEAD -> master) [Sergio Gómez]
| * 9c6ac06 2013-06-16 | Merge branch 'master' into hola (hola) [Sergio Gómez]
| | \
| | /
| / |
* | c3e65d0 2013-06-16 | Añadido README.md [Sergio Gómez]
| * 9862f33 2013-06-16 | hola usa la clase HolaMundo [Sergio Gómez]
| * 6932156 2013-06-16 | Añadida la clase HolaMundo [Sergio Gómez]
| /
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio Gómez]
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto (tag: v1) [Sergio Gómez]
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta) [Sergio Gómez]
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]
```

Volvemos a la rama hola y fusionamos. Recuerda que hemos modificado hola.php en la rama `master` y ahora hay líneas distintas dentro de ese fichero en cada una de las ramas.

```
$ git checkout hola
Switched to branch 'hola'

$ git merge master
Auto-merging lib/hola.php
CONFLICT (content): Merge conflict in lib/hola.php
Automatic merge failed; fix conflicts and then commit the result.
```

Si editamos nuestro archivo `lib/hola.php` obtendremos algo similar a esto:

```
<?php
// Autor: Sergio Gómez <sergio@uco.es>
<<<<<< HEAD
// El nombre por defecto es Mundo
require('HolaMundo.php');

$nombre = isset($argv[1]) ? $argv[1] : "Mundo";
print new HolaMundo($nombre);
=====
print "Introduce tu nombre:";
```

```
$nombre = trim(fgets(STDIN));
@print "Hola, {$nombre}\n";
>>>>>> master
?>
```

La primera parte marca el código que estaba en la rama donde trabajábamos (HEAD), que era la rama `hola` y la parte final el código desde donde fusionábamos, que era `master`. Resolvemos el conflicto, dejando el archivo como sigue:

```
<?php
// Autor: Sergio Gómez <sergio@uco.es>
require('HolaMundo.php');

print "Introduce tu nombre:";
$nombre = trim(fgets(STDIN));
print new HolaMundo($nombre);
?>
```

Y resolvemos el conflicto confirmando los cambios:

```
$ git add lib/hola.php
$ git commit -m "Solucionado el conflicto al fusionar con la rama master"
[hola a36af04] Solucionado el conflicto al fusionar con la rama master
```

## Rebasing vs Merging

Rebasing es otra técnica para fusionar distinta a merge y usa la orden `git rebase`. Vamos a dejar nuestro proyecto como estaba antes del fusionado. Para ello necesitamos anotar el hash anterior al de la acción de *merge*. El que tiene la anotación *"hola usa la clase HolaMundo"*.

Para ello podemos usar la orden `git reset` que nos permite mover HEAD donde queramos.

```
$ git checkout hola
Switched to branch 'hola'
$ git hist
* a36af04 2013-06-16 | Solucionado el conflicto al fusionar con la rama
master (HEAD -> hola) [Sergio Gómez]
|\
| * 9c85275 2013-06-16 | Programa interactivo (master) [Sergio Gómez]
* | 9c6ac06 2013-06-16 | Merge branch 'master' into hola [Sergio Gómez]
|\|
| * c3e65d0 2013-06-16 | Añadido README.md [Sergio Gómez]
* | 9862f33 2013-06-16 | hola usa la clase HolaMundo [Sergio Gómez]
* | 6932156 2013-06-16 | Añadida la clase HolaMundo [Sergio Gómez]
|/
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio Gómez]
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto
(tag: v1) [Sergio Gómez]
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta)
[Sergio Gómez]
```

```
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]
$ git reset --hard 9862f33
HEAD is now at 9862f33 hola usa la clase HolaMundo
```

Y nuestro estado será:

```
$ git hist --all
* 9862f33 2013-06-16 | hola usa la clase HolaMundo (HEAD, hola) [Sergio Gómez]
* 6932156 2013-06-16 | Añadida la clase HolaMundo [Sergio Gómez]
| * 9c85275 2013-06-16 | Programa interactivo (master) [Sergio Gómez]
| * c3e65d0 2013-06-16 | Añadido README.md [Sergio Gómez]
|/
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio Gómez]
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto
(tag: v1) [Sergio Gómez]
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta)
[Sergio Gómez]
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]
```

Hemos deseado todos los *merge* y nuestro árbol está "*limpio*". Vamos a probar ahora a hacer un rebase. Continuamos en la rama `hola` y ejecutamos lo siguiente:

```
$ git rebase master
Auto-merging lib/hola.php
CONFLICT (content): Merge conflict in lib/hola.php
error: could not apply 9862f33... hola usa la clase HolaMundo
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase
--abort".
Could not apply 9862f33... hola usa la clase HolaMundo
```

El conflicto, por supuesto, se sigue dando. Resolvemos guardando el archivo `hola.php` como en los casos anteriores:

```
<?php
// Autor: Sergio Gómez <sergio@uco.es>
require('HolaMundo.php');

print "Introduce tu nombre:";
$nombre = trim(fgets(STDIN));
print new HolaMundo($nombre);
?>
```

Añadimos los cambios en *staging* y en esta ocasión, y tal como nos indicaba en el mensaje anterior, no tenemos que hacer `git commit` sino continuar con el *rebase*:

```
$ git add lib/hola.php
$ git status
interactive rebase in progress; onto 269eaca
Last commands done (2 commands done):
    pick 4e0f425 Añadida clase HolaMundo
    pick 9862f33 hola usa la clase HolaMundo
No commands remaining.
You are currently rebasing branch 'hola' on '8ca56f7'.
    (all conflicts fixed: run "git rebase --continue")

Changes to be committed:
    (use "git restore --staged <file>..." to unstage)
        modified:   lib/hola.php

$ git rebase --continue
[detached HEAD 9862f33] hola usa la clase HolaMundo
1 file changed, 3 insertions(+), 1 deletion(-)
Successfully rebased and updated refs/heads/hola.
```

Y ahora vemos que nuestro árbol tiene un aspecto distinto, mucho más limpio:

```
$ git hist --all
* 9862f33 2013-06-16 | hola usa la clase HolaMundo (HEAD -> hola) [Sergio Gómez]
* 6932156 2013-06-16 | Añadida la clase HolaMundo [Sergio Gómez]
* 9c85275 2013-06-16 | Programa interactivo (master) [Sergio Gómez]
* c3e65d0 2013-06-16 | Añadido README.md [Sergio Gómez]
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio Gómez]
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto (tag: v1) [Sergio Gómez]
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta) [Sergio Gómez]
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]
```

Lo que hace rebase es volver a aplicar todos los cambios a la rama máster, desde su nodo más reciente. Eso significa que se modifica el orden o la historia de creación de los cambios. Por eso rebase no debe usarse si el orden es importante o si la rama es compartida.

## Mezclando con la rama master

Ya hemos terminado de implementar los cambios en nuestra rama secundaria y es hora de llevar los cambios a la rama principal. Usamos `git merge` para hacer una fusión normal:

```
$ git checkout master
Switched to branch 'master'
$ git merge hola
Updating c3e65d0..491f1d2
Fast-forward
```



```

lib/HolaMundo.php | 16 ++++++
lib/hola.php      |  4 +++-
2 files changed, 19 insertions(+), 1 deletion(-)
create mode 100644 lib/HolaMundo.php
$ git hist --all
* 9862f33 2013-06-16 | hola usa la clase HolaMundo (HEAD -> master, hola)
[Sergio Gómez]
* 6932156 2013-06-16 | Añadida la clase HolaMundo [Sergio Gómez]
* 9c85275 2013-06-16 | Programa interactivo [Sergio Gómez]
* c3e65d0 2013-06-16 | Añadido README.md [Sergio Gómez]
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio
Gómez]
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto
(tag: v1) [Sergio Gómez]
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta)
[Sergio Gómez]
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]

```

Vemos que indica que el tipo de fusión es *fast-forward*. Este tipo de fusión tiene el problema que no deja rastro de la fusión, por eso suele ser recomendable usar el parámetro `--no-ff` para que quede constancia siempre de que se ha fusionado una rama con otra.

Vamos a volver a probar ahora sin hacer *fast-forward*. Reseteamos *master* al estado "*Programa interactivo*".

```

$ git reset --hard 9c85275
$ git hist
* 9c85275 2013-06-16 | Programa interactivo (master) [Sergio Gómez]
* c3e65d0 2013-06-16 | Añadido README.md [Sergio Gómez]
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio Gómez]
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto
(tag: v1) [Sergio Gómez]
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta)
[Sergio Gómez]
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]

```

Vemos que estamos como en el final de la sección anterior, así que ahora mezclamos:

```

$ git merge -m "Aplicando los cambios de la rama hola" --no-ff hola
Merge made by the 'recursive' strategy.
 lib/HolaMundo.php | 16 ++++++
 lib/hola.php      |  4 +++-
2 files changed, 19 insertions(+), 1 deletion(-)
create mode 100644 lib/HolaMundo.php
$ git hist --all
* 2eab8ca 2013-06-16 | Aplicando los cambios de la rama hola (HEAD ->
master) [Sergio Gomez]
*\
| * 9862f33 2013-06-16 | hola usa la clase HolaMundo (hola) [Sergio Gómez]
| * 6932156 2013-06-16 | Añadida la clase HolaMundo [Sergio Gómez]

```

```
|/  
* 9c85275 2013-06-16 | Programa interactivo (master) [Sergio Gómez]  
* c3e65d0 2013-06-16 | Añadido README.md [Sergio Gómez]  
* 81c6e93 2013-06-16 | Movido hola.php a lib [Sergio Gómez]  
* 96a39df 2013-06-16 | Añadido el autor del programa y su email [Sergio Gómez]  
* fd4da94 2013-06-16 | Se añade un comentario al cambio del valor por defecto  
(tag: v1) [Sergio Gómez]  
* 3283e0d 2013-06-16 | Se añade un parámetro por defecto (tag: v1-beta)  
[Sergio Gómez]  
* efc252e 2013-06-16 | Parametrización del programa [Sergio Gómez]  
* e19f2c1 2013-06-16 | Creación del proyecto [Sergio Gómez]
```

En la siguiente imagen se puede ver la diferencia:

