

Las ramas en GIT

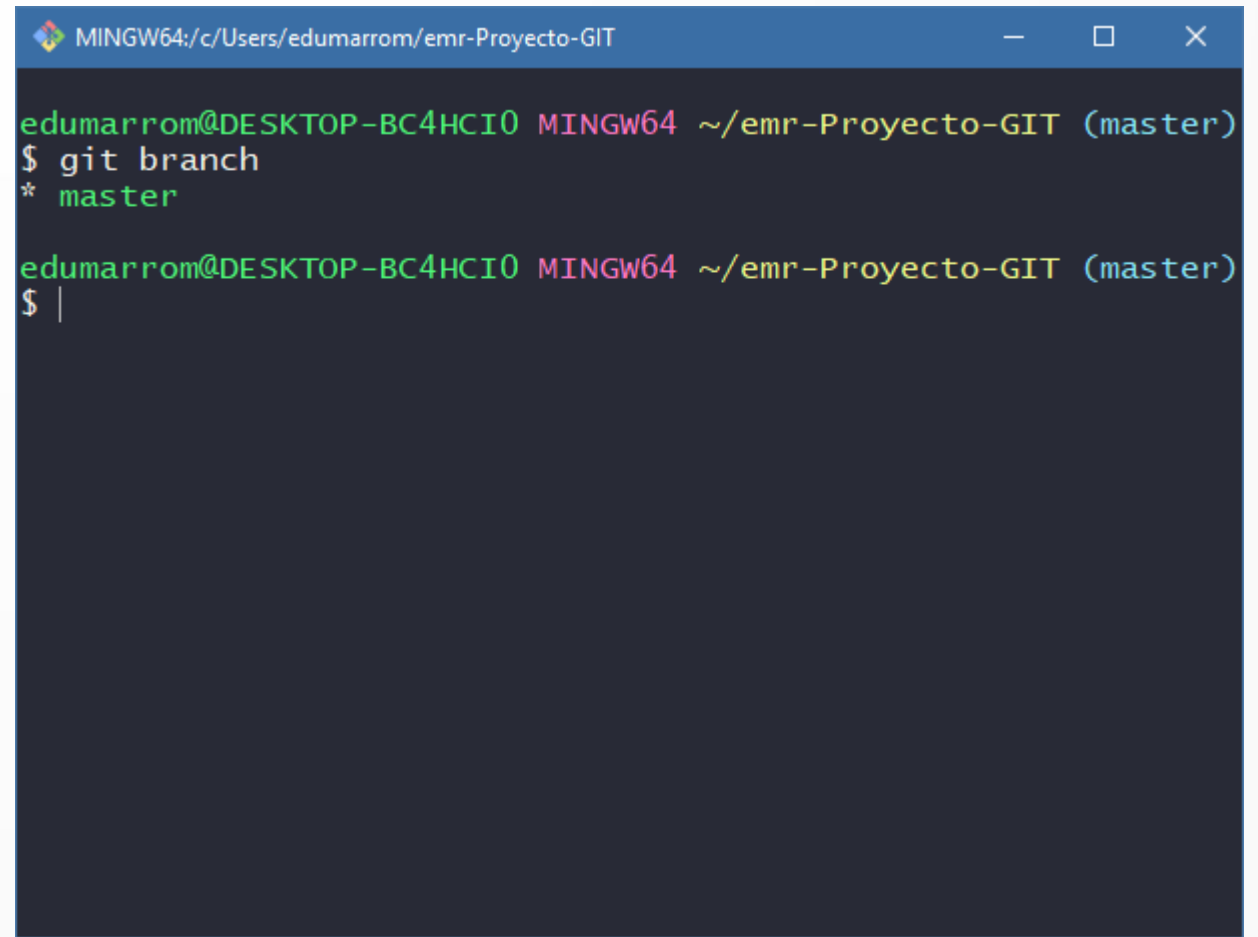
Entornos de Desarrollo
Eduardo Martínez Romero

1 DAW
2020-2021

Introducción a las ramas

1

Las ramas (*branches*) son versiones aisladas de un repositorio que permiten experimentar con él **sin afectar a otras ramas** del mismo. Por defecto, todo repositorio dispone de una rama, llamada *master*, desde la que pueden crearse otras ramas para diversificar distintos desarrollos dentro de un mismo proyecto.

A screenshot of a Windows terminal window titled 'MINGW64: c:/Users/edumarrom/emr-Proyecto-GIT'. The prompt is 'edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)'. The command '\$ git branch' has been entered, and the output is '* master'. The prompt is now '\$ |' on a new line.

```
MINGW64: c:/Users/edumarrom/emr-Proyecto-GIT
edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git branch
* master
edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ |
```

Creando una rama

| 2 - 3

```
MINGW64:/c/Users/edumarrom/emr-Proyecto-GIT
edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git branch
* master

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git branch emr

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git branch
  emr
* master

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git checkout emr
Switched to branch 'emr'

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$
```

Mediante la orden `git branch` podemos **listar las ramas disponibles** de nuestro repositorio. Si añadimos un término después, **creará una nueva rama** con el término que hemos usado.

Si usamos la orden `git checkout [rama]` **cambiaremos** entre las ramas existentes.

Trabajando con ramas

4

En nuestra nueva rama crearemos un fichero con nuestras iniciales, que contenga nuestro nombre. Añadimos el fichero al área de preparación y verificamos su estado. Confirmamos la nueva versión de nuestra rama.

```
MINGW64:/c/Users/edumarrom/emr-Proyecto-GIT
edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ nano emr.txt

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ git status --short
?? emr.txt

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ git add emr.txt
warning: LF will be replaced by CRLF in emr.txt.
The file will have its original line endings in your working
directory

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ git status --short
A  emr.txt

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ git commit -m "rama_emr_texto"
[emr 6d0ca8d] rama_emr_texto
1 file changed, 1 insertion(+)
create mode 100644 emr.txt

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ |
```

Trabajando con ramas

4

```
MINGW64:/c:/Users/edumarrom/emr-Proyecto-GIT
edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ git log --oneline
6d0ca8d (HEAD -> emr) rama_emr_texto
86d3259 (master) emr_Cuarta_version_index.html
3070aee emr-Tercera_version_Imagenes
39544e0 "emr_Segunda_version_JS"
89e7351 "emr_Primer_version_CSS"

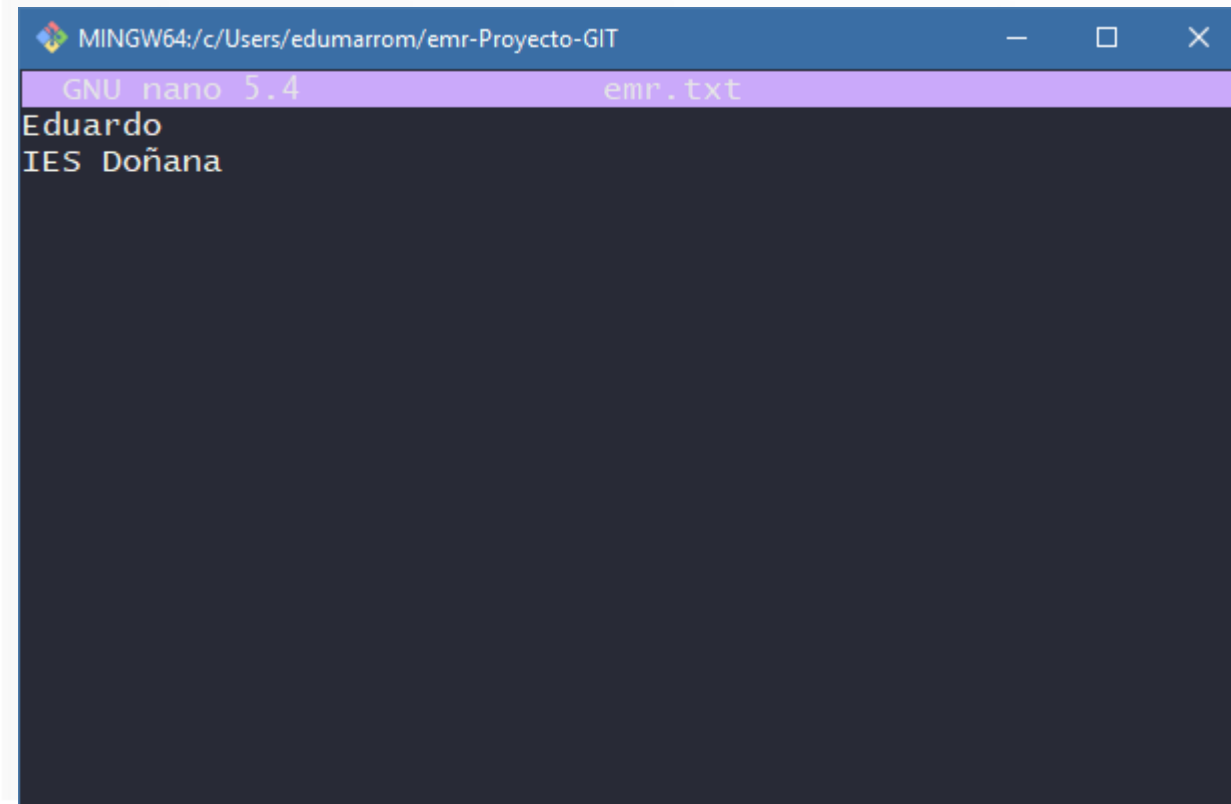
edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$
```

Si revisamos el historial de versiones, podremos observar que **se especifica** cuál es el nombre (entre paréntesis) de **la última rama activa** en la última versión, **antes de cambiar de rama**. Al cambiar de la rama *master* a *emr*, ahora HEAD apunta a *emr*.

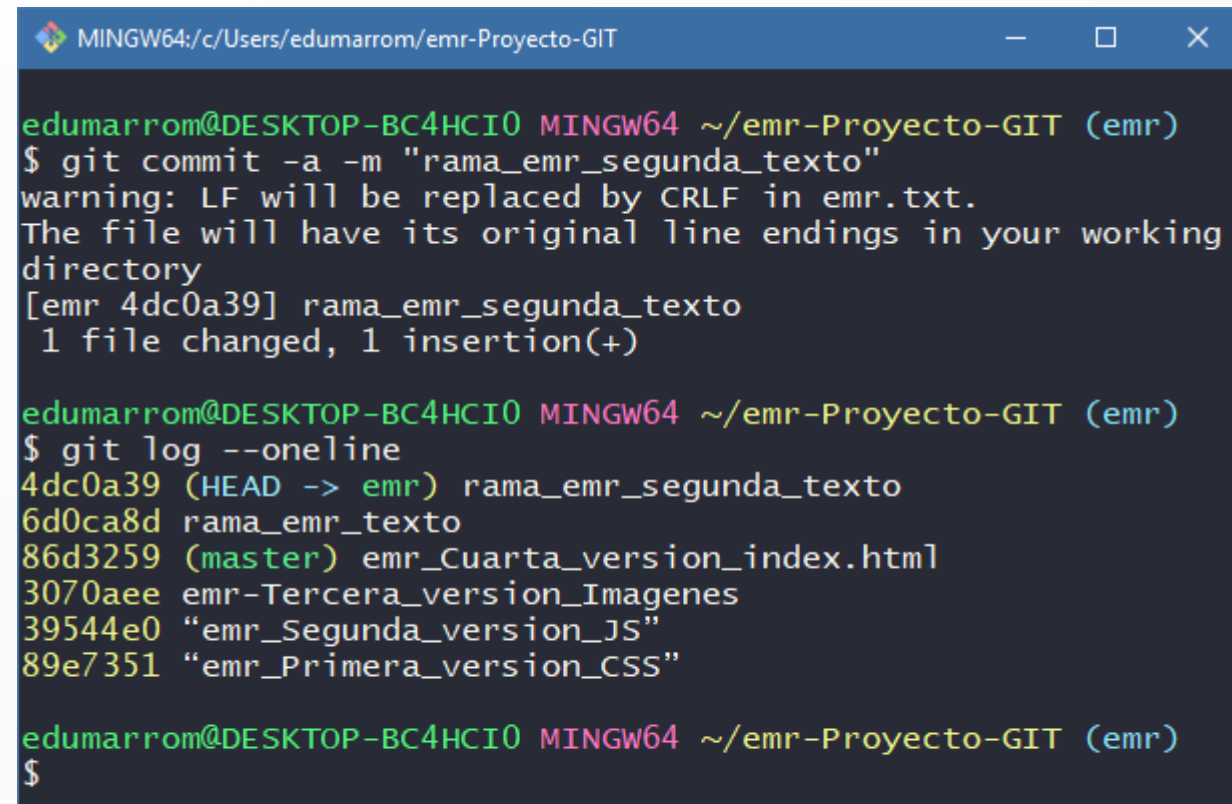
Trabajando con ramas

5

Vamos a modificar el contenido del Revisamos de nuevo el historial de archivo *emr.txt*, añadiendo “*IES Doñana*”. versiones para confirmar que los cambios Preparamos el fichero y creamos una se han realizado correctamente. nueva versión.



```
MINGW64:/c/Users/edumarrom/emr-Proyecto-GIT
GNU nano 5.4 emr.txt
Eduardo
IES Doñana
```



```
MINGW64:/c/Users/edumarrom/emr-Proyecto-GIT

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (emr)
$ git commit -a -m "rama_emr_segunda_texto"
warning: LF will be replaced by CRLF in emr.txt.
The file will have its original line endings in your working
directory
[emr 4dc0a39] rama_emr_segunda_texto
1 file changed, 1 insertion(+)

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (emr)
$ git log --oneline
4dc0a39 (HEAD -> emr) rama_emr_segunda_texto
6d0ca8d rama_emr_texto
86d3259 (master) emr_Cuarta_version_index.html
3070aee emr-Tercera_version_Imagenes
39544e0 "emr_Segunda_version_JS"
89e7351 "emr_Primer_version_CSS"

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (emr)
$
```

Trabajando con ramas

6

Vamos a volver a la rama *master*. En ella, crearemos un nuevo de texto, llamado *elquijote.txt*, con el comienzo del libro. Comprobamos el estado, preparamos el fichero y creamos una nueva versión.

Prestar atención al historial, pues **los cambios realizados en la rama *emr* no aparecen** (al menos en la versión reducida del historial).

```
MINGW64:/c:/Users/edumarrom/emr-Proyecto-GIT

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (emr)
$ git checkout master
Switched to branch 'master'

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (master)
$ nano elquijote.txt

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (master)
$ git status --short
?? elquijote.txt

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (master)
$ git add elquijote.txt
warning: LF will be replaced by CRLF in elquijote.txt.
The file will have its original line endings in your working
directory

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (master)
$ git commit -m "emr_elquijote"
[master e8ddd40] emr_elquijote
1 file changed, 1 insertion(+)
create mode 100644 elquijote.txt

edumarrom@DESKTOP-BC4HCIO MINGW64 ~/emr-Proyecto-GIT (master)
$ git log --oneline
e8ddd40 (HEAD -> master) emr_elquijote
86d3259 emr_Cuarta_version_index.html
3070aee emr-Tercera_version_Imagenes
39544e0 "emr_Segunda_version_JS"
89e7351 "emr_Primer_version_CSS"
```

Trabajando con ramas

7

Ahora modificamos el fichero *quijote.txt* terminando la frase. Verificamos su estado y confirmamos una nueva versión. El historial ahora nos refleja los cambios.

```
MINGW64:/c/Users/edumarrom/emr-Proyecto-GIT
GNU nano 5.4 elquijote.txt Modified
En un lugar de la Mancha de cuyo nombre no quiero acordarme|
```

```
MINGW64:/c/Users/edumarrom/emr-Proyecto-GIT
edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ nano elquijote.txt

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git status --short
M elquijote.txt

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git commit -a -m "emr_elquijote2"
warning: LF will be replaced by CRLF in elquijote.txt.
The file will have its original line endings in your working directory
[master 066a1f7] emr_elquijote2
1 file changed, 2 insertions(+), 1 deletion(-)

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git log --oneline
066a1f7 (HEAD -> master) emr_elquijote2
e8ddd40 emr_elquijote
86d3259 emr_Cuarta_version_index.html
3070aee emr-Tercera_version_Imagenes
39544e0 "emr_Segunda_version_JS"
89e7351 "emr_Primer_version_CSS"
```


Fusionar ramas

| 8 - 9

```
MINGW64:/c:/Users/edumarrom/emr-Proyecto-GIT
edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ ls
css/  elquijote.txt  imagenes/  index.html  js/

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git checkout emr
Switched to branch 'emr'

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ ls
css/  emr.txt  imagenes/  index.html  js/

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (emr)
$ git checkout master
Switched to branch 'master'

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git merge emr
Merge made by the 'recursive' strategy.
  emr.txt | 2 ++
  1 file changed, 2 insertions(+)
  create mode 100644 emr.txt

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ |
```

Si revisamos el contenido de nuestro repositorio, observaremos claramente como **su contenido difiere dependiendo de en que rama nos encontramos**.

Desde la rama *master* **fusionaremos ambas ramas**, por medio de la orden `git merge [rama]`. En mi caso, mediante el editor Vim, **añadiremos un comentario especificando el motivo** de la fusión. Para finalizar, pasamos al modo comando pulsando la tecla *Esc*, y finalizamos guardando y saliendo del editor con “`:wq`”.

Modificando versiones

9

Si revisamos de nuevo el contenido de nuestro repositorio, podremos ver que ambos ficheros ahora se encuentran en nuestro proyecto. En el historial, la fusión **aparecerá** con su comentario **como la última modificación realizada** sobre el proyecto.

```
MINGW64:/c/Users/edumarrom/emr-Proyecto-GIT

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ ls
css/  elquijote.txt  emr.txt  imagenes/  index.html  js/

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ git log --oneline
9acc045 (HEAD -> master) pruebas con git
066a1f7 emr_elquijote2
e8ddd40 emr_elquijote
4dc0a39 (emr) rama_emr_segunda_texto
6d0ca8d rama_emr_texto
86d3259 emr_Cuarta_version_index.html
3070aee emr-Tercera_version_Imagenes
39544e0 "emr_Segunda_version_JS"
89e7351 "emr_Primer_version_CSS"

edumarrom@DESKTOP-BC4HCI0 MINGW64 ~/emr-Proyecto-GIT (master)
$ |
```



git

Gracias a las ramas, podremos añadir y probar nuevas funciones en nuestros proyectos, corregir errores e incluso generar variantes a conveniencia. Todo ello sin necesidad de perder las ventajas de un sistema de control de versiones.