

Git

Taller de introducción

Ejercicio



git

Git. Taller de introducción. Ejercicio

Desarrollado por Jesús Amieiro Becerra

Este archivo se encuentra bajo una licencia [Creative Commons Reconocimiento-CompartirIgual \(CC BY-SA\)](https://creativecommons.org/licenses/by-sa/4.0/). Se permite el uso comercial de la obra y de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.



Versión actual: 09/10/2014

Puedes obtener la última versión disponible de este documento en <http://www.jesusamieiro.com/docs/>

Índice de contenido

1	Presentación.....	5
2	Clonando un repositorio existente.....	5
3	Inicialización a partir de un proyecto existente.....	6
4	Inicialización a partir de un proyecto nuevo.....	8
5	Añadir un primer archivo.....	9
6	Primer commit.....	10
7	Añadiendo más archivos.....	10
8	Editando archivos.....	13
9	Historial del repositorio.....	17
9.1	Otros filtros de git log.....	17
10	Cambios en los diferentes commits.....	18
11	Ramas. Creación.....	21
12	Ramas. Listado.....	23
13	Ramas. Cambio de rama.....	24
14	Ramas. Trabajando con ellas.....	24
15	Rama. Creación y cambio.....	26
16	Renombrar una rama.....	29
17	Borrar una rama.....	30
18	Ramas. Fusión.....	30
19	Conflictos en la fusión entre ramas.....	35
19.1	Abortar la fusión.....	36
19.2	Resolver el problema de forma manual.....	37
19.3	Visualizando gráficamente las ramas.....	39
20	Mostrar información de un objeto.....	40
21	Comparando commits.....	40
22	Etiquetas.....	40
23	Colaborando en un servidor remoto: remote, fetch, push, pull.....	41
23.1	Empezando un proyecto desde cero.....	41
23.2	Sincronizando un proyecto local existente.....	44
23.3	Colaborando con un segundo usuario.....	49

1 Presentación

Este documento es un guion de trabajo elaborado para un taller de introducción a Git, de tal forma que cualquier alumno pueda seguir las explicaciones y/o repetirlo cuando le interese.

2 Clonando un repositorio existente

```
git clone URL [directorio]
```

```
cd ~/proyectos/  
git clone https://github.com/jquery/jquery.git
```

```
ls
```

```
jquery
```

```
ls jquery -la
```

```
AUTHORS.txt  
bower.json  
.bowerrc  
build  
CONTRIBUTING.md  
.editorconfig  
.git  
.gitattributes  
.gitignore  
Gruntfile.js  
.jscs.json  
.jshintignore  
.jshintrc  
.mailmap  
MIT-LICENSE.txt  
.npmignore
```

```
package.json
README.md
src
test
.travis.yml
```

```
cd jquery
git log
```

```
git log --format='%Cred%h%Creset -%C(yellow)%d%Creset %s
%CGreen(%cr)' --abbrev-commit --date=relative -10
```

3 Inicialización a partir de un proyecto existente

Lo primero que hago es crear un directorio de trabajo, que será la zona de trabajo del proyecto que voy a controlar con Git.

```
cd ~/proyectos/
mkdir proyecto_existente
cd proyecto_existente
```

Compruebo el contenido del directorio

```
ls -la
```

Está vacío. Añado dos archivos

```
touch archivo_a.txt
touch archivo_b.txt
```

Compruebo el contenido del directorio

```
ls -la
```

```
archivo_a.txt
archivo_b.txt
```

Inicializo el repositorio.

```
git init
```

```
Initialized empty Git repository in
/home/fontelearn/proyectos/proyecto_existente/.git/
```

Compruebo el contenido y el estado del repositorio

```
ls -la
```

```
archivo_a.txt
archivo_b.txt
.git
```

Hay un directorio llamado ".git", donde Git guarda toda la información relativa al repositorio actual.

```
.git/
├─ branches
├─ config
├─ description
├─ HEAD
├─ hooks
│  ├─ applypatch-msg.sample
│  ├─ commit-msg.sample
│  ├─ post-update.sample
│  ├─ pre-applypatch.sample
│  ├─ pre-commit.sample
│  ├─ prepare-commit-msg.sample
│  ├─ pre-rebase.sample
│  └─ update.sample
├─ info
│  └─ exclude
├─ objects
│  ├─ info
│  └─ pack
└─ refs
    ├─ heads
    └─ tags
```

Compruebo el estado del repositorio

```
git status
```

```
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       archivo_a.txt
#       archivo_b.txt
nothing added to commit but untracked files present (use "git add" to track)
```

4 Inicialización a partir de un proyecto nuevo

```
mkdir ~/proyectos/proyecto_prueba/
cd ~/proyectos/proyecto_prueba/
```

Compruebo el estado del repositorio

```
git status
```

No he inicializado el repositorio

```
fatal: Not a git repository (or any parent up to mount point /home)
Stopping at filesystem boundary (GIT_DISCOVERY_ACROSS_FILESYSTEM not set).
```

```
ls -la
```

```
.
..
```

Inicializo el repositorio

```
git init
```

```
Initialized empty Git repository in
```

```
/home/fontelearn/proyectos/proyecto_prueba/.git/
```

```
ls -la
```

Aparece un nuevo directorio ".git"

```
.  
..  
.git
```

similar al generado en el apartado anterior.

5 Añadir un primer archivo

```
touch archivo_a.txt
```

```
ls -la
```

```
archivo_a.txt  
.git
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
archivo_a.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

Pongo al archivo archivo_a.txt bajo seguimiento y lo paso a la zona de preparación


```
git add archivo_a.txt
```

0

```
git add .
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
Initial commit
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   archivo_a.txt
```

6 Primer commit

Ejecuto el primer commit

```
git commit -m "Añado el archivo_a.txt vacío"
```

```
[master (root-commit) bbc294f] Añado el archivo_a.txt vacío
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
create mode 100644 archivo_a.txt
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
nothing to commit, working directory clean
```

7 Añadiendo más archivos

Creo 2 archivos vacíos

```
touch archivo_b.txt
```

```
touch archivo_c.txt
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
archivo_b.txt
```

```
archivo_c.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

Pongo el archivo archivo_b.txt bajo seguimiento y lo paso a la zona de preparación

```
git add archivo_b.txt
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
new file:   archivo_b.txt
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
archivo_c.txt
```

Realizo el commit

```
git commit -m "Añado el archivo_b.txt vacío"
```

```
[master a09d278] Añado el archivo_b.txt vacío
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo_b.txt
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

  archivo_c.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Pongo el archivo archivo_c.txt bajo seguimiento y lo paso a la zona de preparación

```
git add .
```

Compruebo el estado del repositorio

```
git status
```

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

  new file:   archivo_c.txt
```

Realizo el commit

```
git commit
```

En el editor añado al inicio del archivo el texto

```
Añado el archivo_c.txt vacío
```

Guardo y salgo del editor

```
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
create mode 100644 archivo_c.txt
```

Veo el historial del repositorio

```
git log --oneline
```

```
e6115ee Añado el archivo_c.txt vacío  
a09d278 Añado el archivo_b.txt vacío  
bbc294f Añado el primer archivo vacío
```

8 Editando archivos

Edito el contenido del archivo "archivo_a.txt"

```
echo "Creo una primera línea en archivo_a.txt" >> archivo_a.txt
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master  
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:   archivo_a.txt  
  
no changes added to commit (use "git add" and/or "git commit -a")
```

Almaceno la instantánea del archivo archivo_a.txt en la zona de preparación

```
git add archivo_a.txt
```

Compruebo el estado del repositorio

```
git status
```

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

```
modified:  archivo_a.txt
```

Edito el contenido de los archivos "archivo_b.txt" y "archivo_c.txt"

```
echo "Creo una primera línea en el archivo_b.txt" >> archivo_b.txt
echo "Creo una primera línea en el archivo_c.txt" >> archivo_c.txt
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
modified:  archivo_a.txt
```

```
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:  archivo_b.txt
```

```
modified:  archivo_c.txt
```

Almaceno la instantánea del archivo archivo_b.txt en la zona de preparación

```
git add archivo_b.txt
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
modified:  archivo_a.txt
```

```
modified:  archivo_b.txt
```

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: *archivo_c.txt*

Realizo el commit

```
git commit -m "Introduzco una línea en los archivos archivo_a.txt y
archivo_b.txt"
```

```
[master 689e5d6] Introduzco una línea en los archivos archivo_a.txt y
archivo_b.txt
```

```
2 files changed, 2 insertions(+)
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

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: **archivo_c.txt**

```
no changes added to commit (use "git add" and/or "git commit -a")
```

Compruebo el historial del repositorio

```
git log --oneline
```

```
689e5d6 Introduzco una línea en los archivos archivo_a.txt y archivo_b.txt
```

```
e6115ee Añado el archivo_c.txt vacío
```

```
a09d278 Añado el archivo_b.txt vacío
```

```
bbc294f Añado el primer archivo vacío
```

Almaceno la instantánea del archivo *archivo_c.txt* en la zona de preparación

```
git add archivo_c.txt
```

Realizo el commit

```
git commit -m "Introduzco una línea en el archivo archivo_c.txt"
```

```
[master b73d0fc] Introduzco una línea en el archivo archivo_c.txt  
1 file changed, 1 insertion(+)
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master  
nothing to commit, working directory clean
```

Compruebo el historial del repositorio

```
git log --oneline
```

```
b73d0fc Introduzco una línea en el archivo archivo_c.txt  
689e5d6 Introduzco una línea en los archivos archivo_a.txt y archivo_b.txt  
e6115ee Añado el archivo_c.txt vacío  
a09d278 Añado el archivo_b.txt vacío  
bbc294f Añado el primer archivo vacío
```

9 Historial del repositorio

Vamos a ver el historial del repositorio

```
git log
```

```
commit b04ff0b1ad2eb7da7117757cf5c859841bae165a
Author: Jesus Amieiro <amieiro@gmail.com>
Date:   Sat Oct 4 16:44:32 2014 +0200
    Añado el tercer archivo

commit 81899189b1d0b52ed459f441d52a72deaa98a3af
Author: Jesus Amieiro <amieiro@gmail.com>
Date:   Sat Oct 4 16:42:52 2014 +0200
    Añado el segundo archivo

commit 76dd76ac83f6156dd42f861623dec7277e36e137
Author: Jesus Amieiro <amieiro@gmail.com>
Date:   Sat Oct 4 16:38:02 2014 +0200
    Añado el primer archivo vacío
```

Para ver de una forma mucho más sencilla el historial del repositorio

```
git log --oneline
```

```
b04ff0b Añado el tercer archivo
8189918 Añado el segundo archivo
76dd76a Añado el primer archivo vacío
```

9.1 Otros filtros de git log

```
git log --oneline -6
git log --oneline -6 --graph
git log --oneline -6 --graph --all
git log --oneline -6 --graph --all --decorate
git log --oneline --since=2014-01-24
git log --oneline --until=2014-10-08
```



```
git log --oneline --author="Jesus"  
git log --oneline --author="Jesus *"  
git log --oneline --grep="gitignore"  
git log --oneline --grep="archivo.txt"  
git log --stat --summary -1
```

10 Cambios en los diferentes commits

Voy a utilizar el comando diff para ver cambios llevados a cabo en los distintos commits.

Añado una línea al archivo "archivo_a.txt".

```
echo "Creo una segunda línea en el archivo_a.txt" >> archivo_a.txt
```

Compruebo el contenido del archivo archivo_a.txt

```
cat archivo_a.txt
```

Compruebo el estado del repositorio

```
git status
```

Ahora utilizamos el comando git diff, que muestra los cambios entre commits, entre un commit y el directorio de trabajo,... En este caso va a mostrar los cambios entre el último commit y los cambios llevados a cabo en el directorio de trabajo.

```
git diff
```

Añado una línea al archivo "archivo_b.txt".

```
echo "Creo una segunda línea en el archivo_b.txt" >> archivo_b.txt
```

Compruebo el contenido del archivo archivo_b.txt

```
cat archivo_b.txt
```

Compruebo el estado del repositorio

```
git status
```

Vemos los cambios llevados a cabo desde el último commit

```
git diff
```

Si queremos ver los cambios en un archivo concreto añadimos el nombre como parámetro.

```
git diff archivo_b.txt
```

A continuación muestro cómo ver los cambios en el índice
Añado el archivo `archivo_b.txt` al índice.

```
git add archivo_b.txt
```

Compruebo el estado del repositorio

```
git status
```

Vuelvo a comprobar las diferencias

```
git diff
```

Pero solo aparecen las diferencias de los archivos que no han sido añadidos a la zona de preparación.

Si quiero ver los cambios entre la zona de preparación y el último commit ejecuto

```
git diff --staged
```

O también usando el comando

```
git diff --cached
```

Si quiero ver las diferencias entre mi zona de trabajo y el último commit ejecuto

```
git diff HEAD
```

Añado el archivo `archivo_a.txt` al índice.

```
git add archivo_a.txt
```

Compruebo el estado del repositorio

```
git status
```

Ambos archivos (`archivo_a.txt` y `archivo_b.txt`) están en la zona de preparación.

Si ejecuto

```
git diff
```

La salida es nula, ya que el contenido de la zona de trabajo y la de la zona de preparación es el mismo. Si ejecuto

```
git diff --staged
```

Realizo el comit

```
git commit -m "Añado líneas en el archivo_a.txt y en el
```

```
archivo_b.txt"
```

Compruebo el estado del repositorio

```
git status
```

Compruebo las diferencias ejecutando

```
git diff
```

```
git diff --staged
```

Ambas salidas son nulas, ya que el contenido de la zona de trabajo, de la zona de preparación y del repositorio es el mismo.

11 Ramas. Creación

Inicializo un proyecto, creo 2 archivos y hago 2 commits.

```
cd ~/proyectos/  
mkdir proyecto_ramas  
cd proyecto_ramas  
git init  
touch archivo_a.txt  
git add .  
git commit -m "Añado el primer archivo"  
touch archivo_b.txt  
git add .  
git commit -m "Añado el segundo archivo"
```

Compruebo el historial del repositorio

```
git log --oneline
```

```
1a9f1c2 Añado el segundo archivo  
eff327b Añado el primer archivo
```

Veo las ramas

```
git branch
```

```
* master
```

```
ls -la .git
```

Hay un archivo HEAD.

```
branches  
COMMIT_EDITMSG  
config  
description  
HEAD
```

```
hooks
index
info
logs
objects
refs
```

Veo su contenido

```
cat .git/HEAD
```

Muestra una referencia hacia /refs/heads/master

```
ref: refs/heads/master
```

Veo lo que hay en el directorio "/refs/heads/"

```
ls -la .git/refs/heads/
```

Hay un archivo master referenciado desde el contenido de HEAD

```
master
```

Veo su contenido

```
cat .git/refs/heads/master
```

Es un SHA-1 del último commit de la rama master

```
1a9f1c256aa7aab787fbc0a0d95b971025a13490
```

Creo una nueva rama

```
git branch pruebas
```

```
git branch
```

```
* master
pruebas
```

El "*" nos indica cuál es la rama activa.

Veo el contenido de .git/refs/heads/

```
ls -la .git/refs/heads/
```

```
master
pruebas
```

Aparece un nuevo archivo, "pruebas".

Veo su contenido

```
cat .git/refs/heads/pruebas
```

```
1a9f1c256aa7aab787fbc0a0d95b971025a13490
```

Veo el contenido del archivo de la rama master

```
cat .git/refs/heads/master
```

```
1a9f1c256aa7aab787fbc0a0d95b971025a13490
```

Contienen lo mismo: al mismo commit.

12 Ramas. Listado

Para mostrar todas las ramas

```
git branch
```

```
* master
pruebas
```

Si estuviéramos trabajando con un servidor remoto, para poder mostrar las ramas de seguimiento usaremos el comando

```
git branch -r
```

Para mostrar todas las ramas

```
git branch -a
```

```
* master
pruebas
```

13 Ramas. Cambio de rama

Para mostrar todas las ramas

```
git branch
```

Vemos que estamos en la rama master

```
* master  
pruebas
```

Veo el contenido del HEAD

```
cat .git/HEAD
```

```
ref: refs/heads/master
```

Para cambiarnos de rama

```
git checkout pruebas
```

```
Switched to branch 'pruebas'
```

Muestro las ramas

```
git branch
```

La activa es pruebas

```
master  
* pruebas
```

Veo el contenido del HEAD

```
cat .git/HEAD
```

```
ref: refs/heads/pruebas
```

14 Ramas. Trabajando con ellas

Me cambio a la rama pruebas

```
git checkout pruebas
```

Añado una línea al archivo "archivo_a.txt"

```
echo "Inserto una línea en el archivo_a.txt" >> archivo_a.txt
```

Realizo el commit

```
git commit -am "Introduzco una línea en el archivo archivo_a.txt"
```

```
[pruebas 117d623] Introduzco una línea en el archivo archivo_a.txt  
1 file changed, 1 insertion(+)
```

Veo el historial

```
git log --oneline
```

```
117d623 Introduzco una línea en el archivo archivo_a.txt  
1a9f1c2 Añado el segundo archivo  
eff327b Añado el primer archivo
```

Veo el contenido del archivo "archivo_a.txt"

```
cat archivo_a.txt
```

```
Inserto una línea en el archivo_a.txt
```

Me cambio a la rama "master"

```
git checkout master
```

Veo el contenido del archivo "archivo_a.txt"

```
cat archivo_a.txt
```

Está vacío.

Veo el historial

```
git log --oneline
```



```
1a9f1c2 Añado el segundo archivo
eff327b Añado el primer archivo
```

Me cambio a la rama pruebas

```
git checkout pruebas
```

Veo el historial

```
git log --oneline
```

```
117d623 Introduzco una línea en el archivo archivo_a.txt
1a9f1c2 Añado el segundo archivo
eff327b Añado el primer archivo
```

Veo el contenido del archivo "*archivo_a.txt*"

```
cat archivo_a.txt
```

```
Inserto una línea en el archivo_a.txt
```

El contenido del espacio de trabajo cambia acorde a nuestros cambios y commits en la rama activa.

15 Rama. Creación y cambio

Para crear una rama a partir de otra, me cambio a la rama de partida

```
git checkout master
```

Y creo la rama

```
git branch [nombre_de_la_rama]
```

Luego me cambio a esa nueva rama

```
git checkout [nombre_de_la_rama]
```

Lo puedo hacer con un único comando

```
git checkout master
git checkout -b experimento
```

```
Switched to a new branch 'experimento'
```

Añado una línea al archivo "archivo_a.txt".

```
echo "Experimento añadiendo una nueva línea al archivo_a.txt en la
rama experimento" >> archivo_a.txt
```

Compruebo el estado del repositorio

```
git status
```

```
# On branch experimento
# 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:   archivo_a.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Realizo el commit

```
git commit -am "Experimentando con una nueva línea en el
archivo_a.txt en la rama experimento"
```

```
[experimento 9618f2a] Experimentando con una nueva línea en el archivo_a.txt
en la rama experimento
1 file changed, 1 insertion(+)
```

Veo el histórico de commits

```
git log --oneline
```

```
9618f2a Experimentando con una nueva línea en el archivo_a.txt en la rama
experimento
1a9f1c2 Añado el segundo archivo
eff327b Añado el primer archivo
```

Veo el contenido del archivo

```
cat archivo_a.txt
```

```
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
```

Me cambio a la rama "master"

```
git checkout master
```

Veo el histórico de commits

```
git log --oneline
```

```
1a9f1c2 Añado el segundo archivo  
eff327b Añado el primer archivo
```

Veo el contenido del archivo "archivo_a.txt"

```
cat archivo_a.txt
```

Está vacío.

Me cambio a la rama "pruebas"

```
git checkout pruebas
```

Veo el histórico de commits

```
git log --oneline
```

```
117d623 Introduzco una línea en el archivo archivo_a.txt  
1a9f1c2 Añado el segundo archivo  
eff327b Añado el primer archivo
```

Tampoco tenemos el último commit: se ha llevado a cabo en la rama "experimento".

Veo el contenido del archivo "archivo_a.txt"

```
cat archivo_a.txt
```

```
Inserto una línea en el archivo_a.txt
1a9f1c2 Añado el segundo archivo
eff327b Añado el primer archivo
```

El contenido es distinto del introducido en la rama "experimento".

Ejecuto

```
git log --oneline --graph --all --decorate
```

```
* 9618f2a (experimento) Experimentando con una nueva línea en el archivo_a.txt
| * 117d623 (HEAD, pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 1a9f1c2 (master) Añado el segundo archivo
* eff327b Añado el primer archivo
```

16 Renombrar una rama

```
git branch -m [nombre_actual] [nombre_nuevo]
```

Veo las ramas

```
git branch
```

```
experimento
master
* pruebas
```

Renombro la rama "experimento"

```
git branch -m experimento intento
```

Veo las ramas

```
git branch
```

```
intento
master
* pruebas
```

17 Borrar una rama

```
git branch -d [nombre_de_la_rama]
```

Creo una rama nueva que borro a continuación

```
git checkout master
git branch rama_temporal
git checkout rama_temporal
git branch -d rama_temporal
```

```
error: Cannot delete the branch 'rama_temporal' which you are currently on.
```

Me cambio a la rama "master"

```
git checkout master
git branch -d rama_temporal
```

```
Deleted branch rama_temporal (was 1a9f1c2).
```

18 Ramas. Fusión

```
git merge [rama_a_fusionar]
```

Veo las ramas

```
git branch
```

```
  intento
* master
  pruebas
```

Veo los commits de las distintas ramas

```
git log --oneline --graph --all --decorate
```

```
* 9618f2a (intento) Experimentando con una nueva línea en el archivo_a.txt
| * 117d623 (pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 1a9f1c2 (HEAD, master) Añado el segundo archivo
* eff327b Añado el primer archivo
```

```
git checkout master
```

```
git merge intento
```

```
Updating 1a9f1c2..9618f2a
Fast-forward
 archivo_a.txt | 1 +
 1 file changed, 1 insertion(+)
```

Se ha hecho un "*fast-forward*"

Veo los commits de las distintas ramas

```
git log --oneline --graph --all --decorate
```

```
* 9618f2a (HEAD, master, intento) Experimentando con una nueva línea en el
archivo_a.txt en la rama experimento
| * 117d623 (pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 1a9f1c2 Añado el segundo archivo
* eff327b Añado el primer archivo
```

Cambio a la rama "master"

```
git checkout master
```

Veo contenido del archivo "archivo_a.txt"

```
cat archivo_a.txt
```

```
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
```

Realizo un cambio en la rama "*master*" y otro en la rama "*intento*"

```
echo "Añado una segunda línea al archivo_a.txt en la rama master" >>
archivo_a.txt
```

Hago el commit

```
git commit -am "Añado una segunda línea al archivo_a.txt en la rama
master"
```

```
[master 9229a2f] Añado una segunda línea al archivo_a.txt en la rama master
1 file changed, 1 insertion(+)
```

Rama "*intento*".

```
git checkout intento
echo "Añado una primera línea al archivo_b.txt en la rama intento"
>> archivo_b.txt
git commit -am "Añado una primera línea al archivo_b.txt en la rama
intento"
```

```
[intento 41077ec] Añado una primera línea al archivo_b.txt en la rama intento
1 file changed, 1 insertion(+)
```

Veo los commits de las distintas ramas

```
git log --oneline --graph --all --decorate
```

```
* 41077ec (HEAD, intento) Añado una primera línea al archivo_b.txt en la rama
| * 9229a2f (master) Añado una segunda línea al archivo_a.txt en la rama
|/
* 9618f2a Experimentando con una nueva línea en el archivo_a.txt en la rama
| * 117d623 (pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 1a9f1c2 Añado el segundo archivo
* eff327b Añado el primer archivo
```

```
git checkout master
```

git merge intento

Se abre el editor , ya que va a crear un *commit de fusión* de forma automática con la fusión.

```
Merge branch 'intento'

# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

Dejo el mensaje tal y como está, guardo y salgo del editor.

```
Merge made by the 'recursive' strategy.
 archivo_b.txt | 1 +
 1 file changed, 1 insertion(+)
```

Veo el contenido del archivo "*archivo_b.txt*" en la rama "*master*"

```
git checkout master
cat archivo_b.txt
```

```
Añado una primera línea al archivo_b.txt en la rama intento
```

Veo los commits de las distintas ramas

```
git log --oneline --graph --all --decorate
```

```
* ef17121 (HEAD, master) Merge branch 'intento'
|\
| * 41077ec (intento) Añado una primera línea al archivo_b.txt en la rama
* | 9229a2f Añado una segunda línea al archivo_a.txt en la rama master
|/
* 9618f2a Experimentando con una nueva línea en el archivo_a.txt en la rama
| * 117d623 (pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 1a9f1c2 Añado el segundo archivo
```


* eff327b Añado el primer archivo

19 Conflictos en la fusión entre ramas

Veo el contenido del archivo "archivo_a.txt" en la rama "master"

```
git checkout master
cat archivo_a.txt
```

```
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
Añado una segunda línea al archivo_a.txt en la rama master
```

Veo el contenido del archivo "archivo_a.txt" en la rama "pruebas"

```
git checkout pruebas
cat archivo_a.txt
```

```
Inserto una línea en el archivo_a.txt
```

La primera línea es diferente.

Llevo los cambios de la rama "*pruebas*" a la rama "*master*".

```
git checkout master
git merge pruebas
```

```
Auto-merging archivo_a.txt
CONFLICT (content): Merge conflict in archivo_a.txt
Automatic merge failed; fix conflicts and then commit the result.
```

La fusión falla

Veo el estado del repositorio

```
git status
```

```
# On branch master
# Unmerged paths:
#   (use "git add/rm <file>..." as appropriate to mark resolution)
#
# both modified:   archivo_a.txt
```

```
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Me indica que hay rutas que no están fusionadas y que no se han añadido cambios al commit.

Veo el contenido del archivo en conflicto

```
cat archivo_a.txt
```

```
<<<<<<< HEAD
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
Añado una segunda línea al archivo_a.txt en la rama master
=====
Inserto una línea en el archivo_a.txt
>>>>>>> pruebas
```

Opciones:

- Abortar la fusión.
- Resolver el problema de forma manual y luego finalizar la fusión.
- Utilizar herramientas de fusión.

19.1 Abortar la fusión

```
git merge --abort
```

Y todo vuelve al estado anterior al inicio de la fusión.

Veo el estado del repositorio

```
git status
```

```
# On branch master
nothing to commit (working directory clean)
```

Veo el contenido del archivo "archivo_a.txt"

```
cat archivo_a.txt
```

```
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
```

```
Añado una segunda línea al archivo_a.txt en la rama master
```

Vemos que ninguno de los cambios de la rama "pruebas" ha sido llevado a este archivo en la rama "master".

19.2 Resolver el problema de forma manual

Vuelvo a ejecutar la fusión

```
git checkout master
git merge pruebas
```

```
Auto-merging archivo_a.txt
CONFLICT (content): Merge conflict in archivo_a.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Veo el contenido del archivo en conflicto

```
cat archivo_a.txt
```

```
<<<<<<< HEAD
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
Añado una segunda línea al archivo_a.txt en la rama master
=====
Inserto una línea en el archivo_a.txt
>>>>>>> pruebas
```

Parte en conflicto en la rama activa, la "master", delimitada por

```
<<<<<<< HEAD
```

Y

```
=====
```

Parte en conflicto en la rama "pruebas", delimitada por

```
=====
```

Y

```
>>>>>>> pruebas
```

Edito el archivo, eliminando las marcas que delimitan el problema

```
echo "Resuelvo el conflicto en esta primera línea del archivo_a.txt"
> archivo_a.txt

echo "También modifíco la segunda línea del archivo_a.txt" >>
archivo_a.txt
```

Veo el contenido del archivo en conflicto

```
cat archivo_a.txt
```

```
Resuelvo el conflicto en esta primera línea del archivo_a.txt
También modifíco la segunda línea del archivo_a.txt
```

Veo el estado del repositorio

```
git status
```

```
# On branch master
# Unmerged paths:
#   (use "git add/rm <file>..." as appropriate to mark resolution)
#
# both modified:   archivo_a.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Ejecuto

```
git add archivo_a.txt
```

Veo el estado del repositorio

```
git status
```

```
# On branch master
# Changes to be committed:
#
#   modified:   archivo_a.txt
#
```

Hago el commit

```
git commit -m "Resuelto el conflicto en la línea 1 en el
archivo_a.txt"
```

```
[master 70b7bef] Resuelto el conflicto en la línea 1 en el archivo_a.txt
```

Veo el histórico de commits

```
git log --oneline --graph --all --decorate -6
```

La rama "pruebas" se ha integrado en la rama "master"

```
* 70b7bef (HEAD, master) Resuelto el conflicto en la línea 1 en el
|\
| * 117d623 (pruebas) Introduzco una línea en el archivo archivo_a.txt
* | ef17121 Merge branch 'intento'
|\ \
| * | 41077ec (intento) Añado una primera línea al archivo_b.txt en la rama
* | | 9229a2f Añado una segunda línea al archivo_a.txt en la rama master
|/ /
* | 9618f2a Experimentando con una nueva línea en el archivo_a.txt en la rama
|/
```

19.3 Visualizando gráficamente las ramas

```
git log --oneline -6
git log --oneline -6 --graph
git log --oneline -6 --graph --all
git log --oneline -6 --graph --all --decorate
```

20 Mostrar información de un objeto

```
git branch
git log --oneline
git show HEAD
git show --format=oneline HEAD
git show --oneline HEAD
git show --oneline HEAD^
git show --oneline HEAD^^
git show --oneline HEAD~2
git show --oneline HEAD~6
git show --oneline HEAD~5
git show --oneline $SHA1_commit
```

21 Comparando commits

```
git diff prueba..desarrollo
git diff desarrollo..prueba
git diff prueba^..desarrollo
git diff prueba^..desarrollo^ /// Vacío
git log --oneline -6 --graph --all --decorate
git diff prueba..desarrollo archivo_segundo.txt /// Vacío
git log --oneline
git diff --stat --summary $SHA1_commit_inicial..HEAD // Trabajo realizado
git diff --stat --summary $SHA1_commit_inicial.. .gitignore
git diff $SHA1_commit_inicial.. .gitignore
```

22 Etiquetas

```
git tag
git tag -a v1.0 $SHA1-commit -m "Versión 1.0" //4º último commit
git tag
git show v1.0
git tag -a v1.1. -m "Versión 1.1. Fusionadas las ramas de desarrollo y master"
git tag
git show tag v1.1
git tag -l "v1.*"
git tag -l "v*.1"
```

23 Colaborando en un servidor remoto: remote, fetch, push, pull

23.1 Empezando un proyecto desde cero

Crear un repositorio en Bitbucket con las siguientes características. Botón "Create". Nombre "prueba_html".

```
This is a private repository.  
Allow only private forks.  
Git.  
Issue tracking.  
Wiki.  
Language: HTML/CSS.
```

Creo el directorio

```
mkdir ~/prueba_html/  
cd ~/prueba_html/
```

Inicializo el repositorio

```
git init
```

Veo los repositorios remotos que tengo configurados

```
git remote
```

Salida vacía

Añado un repositorio remoto

```
git remote add origin  
https://amieiro@bitbucket.org/amieiro/prueba_html.git
```

Veo los repositorios remotos que tengo ahora mismo

```
git remote  
git remote -v
```

Creo que primer archivo en local


```
echo "#Mi fichero README" >> README.md
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
README.md
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

Añado el archivo al repositorio y lo coloco en la zona de preparación

```
git add README.md
```

Realizo el commit

```
git commit -m "Commit inicial. Añado el archivo README"
```

```
[master (root-commit) f8c6f11] Commit inicial. Añado el archivo README
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Sincronizo la rama local con el repositorio remoto

```
git push [nombre_remoto][nombre_rama_local]
```

```
git push -u origin master
```

```
Counting objects: 3, done.
```

```
Writing objects: 100% (3/3), 259 bytes | 0 bytes/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0)
To https://amieiro@bitbucket.org/amieiro/prueba_html.git
* [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Veo en Bitbucket estos cambios.

Añado dos archivos más

```
touch index.html
touch style.css
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
Your branch is up-to-date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

  index.html
  style.css

nothing added to commit but untracked files present (use "git add" to track)
```

Sincronizo la rama local con el repositorio remoto

```
git push
```

```
Everything up-to-date
```

Añado los archivos al repositorio y los coloco en la zona de preparación

```
git add .
```

Realizo el commit

```
git commit -m "Añado el index.html y el style.css"
```

```
[master d8cc404] Añado el index.html y el style.css
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
create mode 100644 style.css
```

Sincronizo la rama local con el repositorio remoto

```
git push
```

```
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 313 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://amieiro@bitbucket.org/amieiro/prueba_html.git
f8c6f11..d8cc404 master -> master
```

Veo en Bitbucket estos cambios.

23.2 Sincronizando un proyecto local existente

Creo el directorio

```
mkdir ~/prueba_html_2/
cd ~/prueba_html_2/
```

Inicializo el repositorio

```
git init
```

```
Initialized empty Git repository in /home/amieiro/prueba_html_2/.git/
```

Creo 2 archivos

```
touch index.html
touch style.css
```

Compruebo el estado del repositorio

```
git status
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
index.html
```

```
style.css
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

Añado los archivos al repositorio y los coloco en la zona de preparación

```
git add .
```

Realizo el commit

```
git commit -m "Commit inicial. Añado index.html y style.css"
```

Creo un repositorio en BitBucket. Botón "Create". Nombre "prueba_html_2".

Privado.

```
Allow only private forks.
```

```
Git.
```

```
Issue tracking.
```

```
Wiki.
```

```
Language: HTML/CSS.
```

Veo los repositorios remotos que tengo configurados

```
git remote
```

Añado un repositorio remoto

```
git remote add origin  
https://amieiro@bitbucket.org/amieiro/prueba_html_2.git
```

Veo los repositorios remotos que tengo configurados

```
git remote -v
```

```
origin https://amieiro@bitbucket.org/amieiro/prueba_html_2.git (fetch)  
origin https://amieiro@bitbucket.org/amieiro/prueba_html_2.git (push)
```

Sincronizo el repositorio local con el repositorio remoto

```
git push -u origin master
```

```
Counting objects: 3, done.  
Delta compression using up to 2 threads.  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 253 bytes | 0 bytes/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
To https://amieiro@bitbucket.org/amieiro/prueba_html_2.git  
* [new branch]      master -> master  
Branch master set up to track remote branch master from origin.
```

Compruebo el contenido del archivo de configuración

```
cat .git/config
```

```
[remote "origin"]  
  url = https://amieiro@bitbucket.org/amieiro/prueba_html_2.git  
  fetch = +refs/heads/*:refs/remotes/origin/*  
[branch "master"]  
  remote = origin  
  merge = refs/heads/master
```

Veo en Bitbucket estos cambios.

Añado modificaciones y un nuevo commit

```
echo "#Mi fichero README" >> README.md  
git add README.md  
git commit -m "Añado el archivo README"
```

```
[master 52d13e8] Añado el archivo README
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Sincronizo el repositorio local con el repositorio remoto

```
git push -u origin master
```

```
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 329 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://amieiro@bitbucket.org/amieiro/prueba_html_2.git
    4909473..52d13e8  master -> master
Branch master set up to track remote branch master from origin.
```

Añado modificaciones y un nuevo commit

```
echo "#Una segunda línea comentada" >> README.md
git commit -am "Añado una segunda línea al archivo README.md"
```

```
[master 1768c5b] Añado una segunda línea al archivo README.md
1 file changed, 1 insertion(+)
```

```
git push
```

```
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 377 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://amieiro@bitbucket.org/amieiro/prueba_html_2.git
    52d13e8..1768c5b  master -> master
```

Añado 1 rama a partir de la rama master

```
git branch desarrollo
```

Me cambio a la rama "desarrollo"

```
git checkout desarrollo
```

Añado un archivo en la rama desarrollo, le añado contenido y hago el commit

```
echo "Creo el archivo_1.txt en la rama desarrollo" >> archivo_1.txt  
git add .  
git commit -m "Creo el archivo_1.txt en la rama desarrollo"
```

```
[rama_1 c534304] Creo el archivo_1.txt en la rama 1  
1 file changed, 1 insertion(+)  
create mode 100644 archivo_1.txt
```

Veo las ramas actuales

```
git branch -a
```

```
* desarrollo  
master  
remotes/origin/master
```

Veo el historial de commits

```
git log --oneline
```

Sincronizo el repositorio local con el repositorio remoto

```
git push
```

```
fatal: The current branch rama_1 has no upstream branch.  
To push the current branch and set the remote as upstream, use
```

```
git push --set-upstream origin rama_1
```

Falla

```
git push -u origin desarrollo
```

```
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 398 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://amieiro@bitbucket.org/amieiro/prueba_html_2.git
 * [new branch]      desarrollo -> desarrollo
Branch desarrollo set up to track remote branch desarrollo from origin.
```

Veo en Bitbucket estos cambios.

23.3 Colaborando con un segundo usuario

En Bitbucket añado un segundo usuario. Para ello voy a Share e introduzco el usuario. Le doy permisos de administración.

En el panel de administración del segundo usuario ir a Inbox. Aparece un mensaje como

```
You have been given admin access to amieiro/prueba_html_2
```

USUARIO 2

Simulando el segundo usuario, accedo a su carpeta de usuario

```
mkdir ~/prueba_html_2_usuario2/
cd ~/prueba_html_2_usuario2/
```

Clono el proyecto en la carpeta "prueba_html_2"

```
git clone
https://JesusAmieiro@bitbucket.org/amieiro/prueba_html_2.git
prueba_html_2
```

```
Cloning into 'prueba_html_2'...
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 12 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (12/12), done.
Checking connectivity... done.
```

Accedo al repositorio


```
cd ~/prueba_html_2_usuario2/prueba_html_2/
```

Veo las ramas existentes

```
git branch
```

```
* master
```

```
git branch -a
```

```
* master
remotes/origin/HEAD -> origin/master
remotes/origin/desarrollo
remotes/origin/master
```

Veo la configuración

```
cat .git/config
```

```
[remote "origin"]
  url = https://JesusAmieiro@bitbucket.org/amieiro/prueba_html_2.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
  merge = refs/heads/master
```

Creo la rama local a partir de una rama de seguimiento

```
git branch desarrollo origin/desarrollo
```

```
Branch desarrollo set up to track remote branch desarrollo from origin.
```

Me cambio a esa rama

```
git checkout desarrollo
```

```
Switched to branch 'desarrollo'  
Your branch is up-to-date with 'origin/desarrollo'.
```

Veo las ramas

```
git branch -a
```

```
* desarrollo  
master  
remotes/origin/HEAD -> origin/master  
remotes/origin/desarrollo  
remotes/origin/master
```

Veo la configuración

```
cat .git/config
```

```
[remote "origin"]  
  url = https://JesusAmieiro@bitbucket.org/amieiro/prueba_html_2.git  
  fetch = +refs/heads/*:refs/remotes/origin/*  
[branch "master"]  
  remote = origin  
  merge = refs/heads/master  
[branch "desarrollo"]  
  remote = origin  
  merge = refs/heads/desarrollo
```

Añado un nuevo archivo de CSS en la rama desarrollo, hago el commit y lo subo al servidor.

```
touch style_dev.css  
git add .  
git commit -m "Creo un nuevo archivo de CSS en la rama desarrollo"
```

```
[master b94b8bd] Creo un nuevo archivo de CSS en la rama desarrollo  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 style_dev.css
```

git push

```
Counting objects: 4, done.  
Delta compression using up to 2 threads.  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (2/2), 268 bytes | 0 bytes/s, done.  
Total 2 (delta 1), reused 0 (delta 0)  
To https://JesusAmieiro@bitbucket.org/amieiro/prueba_html_2.git  
f0674ba..0c8b20c desarrollo -> desarrollo
```

Ver el resultado en BitBucket, en los 2 repositorios.
Ahora me cambio al usuario 1 en local

USUARIO 1

git log --oneline master

```
4e5d1ae Añado una segunda línea al archivo README.md  
72d7bca Añado el archivo README  
379c535 Commit inicial. Añado index.html y style.css
```

git log --oneline desarrollo

```
f0674ba Creo el archivo_1.txt en la rama desarrollo  
4e5d1ae Añado una segunda línea al archivo README.md  
72d7bca Añado el archivo README  
379c535 Commit inicial. Añado index.html y style.css
```

Veo el estado del repositorio

git status

```
On branch desarrollo
```

```
Your branch is up-to-date with 'origin/desarrollo'.
```

```
nothing to commit, working directory clean
```

Traigo los cambios del servidor a las ramas de seguimiento

```
git fetch
```

```
remote: Counting objects: 2, done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 2 (delta 1), reused 0 (delta 0)  
Unpacking objects: 100% (2/2), done.  
From https://bitbucket.org/amieiro/prueba_html_2  
    f0674ba..0c8b20c  desarrollo -> origin/desarrollo
```

Veo las diferencias entre la rama de seguimiento y la rama local

```
git diff desarrollo..origin/desarrollo
```

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

Veo los commits de la rama local "desarrollo".

```
git log --oneline desarrollo
```

```
f0674ba Creo el archivo_1.txt en la rama desarrollo  
4e5d1ae Añado una segunda línea al archivo README.md  
72d7bca Añado el archivo README  
379c535 Commit inicial. Añado index.html y style.css
```

Veo los commits de la rama de seguimiento "origin/desarrollo".

```
git log --oneline origin/desarrollo
```

```
0c8b20c Creo un nuevo archivo de CSS en la rama desarrollo  
f0674ba Creo el archivo_1.txt en la rama desarrollo
```

```
4e5d1ae Añado una segunda línea al archivo README.md
72d7bca Añado el archivo README
379c535 Commit inicial. Añado index.html y style.css
```

Fusiono las ramas

```
git checkout desarrollo
```

```
Already on 'desarrollo'
Your branch is behind 'origin/desarrollo' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

```
git merge origin/desarrollo
```

```
Updating f0674ba..0c8b20c
Fast-forward
 style_dev.css | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 style_dev.css
```

Si vuelvo a ver las diferencias entre las ramas

```
git diff desarrollo..origin/desarrollo
```

No existen.

Si veo el historial de commits de la rama local de desarrollo

```
git log --oneline desarrollo
```

```
0c8b20c Creo un nuevo archivo de CSS en la rama desarrollo
f0674ba Creo el archivo_1.txt en la rama desarrollo
4e5d1ae Añado una segunda línea al archivo README.md
72d7bca Añado el archivo README
```

```
379c535 Commit inicial. Añado index.html y style.css
```

Aparece el commit creado por el otro usuario.

```
git fetch + git merge = git pull
```