Veamos como crear un repositorio **git** para trabajar con nuestros archivos de código fuente, y las operaciones básicas de las que disponemos para interactuar con el repositorio.

**Inicializar un repositorio**

1. # Creamos un directorio y un archivo
2. mkdir helloworld
3. cd helloworld
4. echo "Hello World" > README
6. # Inicializamos el repositorio
7. git init
8. Initialized empty Git repository in /home/dave/helloworld/.git/

**Añadir archivos al repositorio y realizar el primer commit**

1. git add README
2. git commit -d "First commit"

**Enlazar con un repositorio remoto**

Imaginaros ahora que queremos que nuestro repositorio local recien creado se comunique con un repositorio remoto. Definiremos para ello un **remote**, con lo cual podremos empezar a realizar comandos **pull/fecth**, y si tenemos permiso, también **push**.

1. git remote add origin git@gitlab.micronautas.com:testing.git
2. git push -u origin master

**Obtener un repositorio remoto**

Normalmente no partimos de cero en nuestro repositorio, sino que ya existe (por ejemplo en **github.com**). Si quisiéramos obtener una copia local de ese repositorio remoto, estableciendo además automáticamente el **remote** a ese repositorio, utilizaremos el siguiente comando:

1. $ git clone git@github.com:tombatossals/git-puesto-en-practica.git
2. Cloning into 'git-puesto-en-practica'...
3. remote: Counting objects: 293, done.
4. remote: Compressing objects: 100% (258/258), done.
5. remote: Total 293 (delta 123), reused 197 (delta 27)
6. Receiving objects: 100% (293/293), 3.13 MiB | 234 KiB/s, done.
7. Resolving deltas: 100% (123/123), done.
8. $ cd git-puesto-en-practica/
9. $ git branch
10. \* master
11. $ git remote -v
12. origin git@github.com:tombatossals/git-puesto-en-practica.git (fetch)
13. origin git@github.com:tombatossals/git-puesto-en-practica.git (push)

**Permisos de acceso a un repositorio remoto**

Los proyectos de software libre suelen tener permisos de lectura sin autenticación (p.e. en [github](http://github.com/)), pero para obtener permisos de escritura deberemos autenticarnos de alguna manera. Primero que nada, deberemos configurar git para que envíe nues

$ git config --global user.name "Peter Pan"

$ git config --global user.email "peter.pan@nuncajamas.com"

**Listando los últimos cambios del repositorio**

Listemos ahora los últimos cambios que se han producido en el repositorio, es decir, los últimos **commits** que han realizado los desarrolladores del repositorio:

1. $ git log --oneline
2. fbcf1a6 Optimized for ipad
3. 9c0a361 Some minor changes
4. 7d8ccd7 Renamed project
5. 81716a8 Some cosmetic changes, and finished the second presentation.
6. fde35e7 Lot of changes
7. e7286e0 Working as team slides already finished
8. b0c3a69 More slides
9. ...

**Obtener los últimos cambios del repositorio remoto**

Veamos ahora cómo obtener todos los últimos cambios del repositorio remoto haciendo un **fetch**.

1. $ git branch -a
2. \* master
3. remotes/origin/master
4. $ git fetch
5. remote: Counting objects: 13, done.
6. remote: Compressing objects: 100% (5/5), done.
7. remote: Total 8 (delta 3), reused 8 (delta 3)
8. Unpacking objects: 100% (8/8), done.
9. From github.com:tombatossals/unimajors
10. 0486878..d0f88cc master -> origin/master
11. $ git diff origin/master
12. diff --git a/2011-iim/sesion01/images/alineacion.jpg b/2011-iim/sesion01/images/alineacion.jpg
13. deleted file mode 100644
14. index 93d68ae..0000000
15. Binary files a/2011-iim/sesion01/images/alineacion.jpg and /dev/null differ
16. diff --git a/2011-iim/sesion01/images/espaciado.jpg b/2011-iim/sesion01/images/espaciado.jpg
17. deleted file mode 100644
18. index c902e4a..0000000
19. Binary files a/2011-iim/sesion01/images/espaciado.jpg and /dev/null differ
20. diff --git a/2011-iim/sesion01/index.html b/2011-iim/sesion01/index.html
21. index 5210ff5..b1d02f8 100755
22. --- a/2011-iim/sesion01/index.html
23. +++ b/2011-iim/sesion01/index.html
24. @@ -287,14 +287,14 @@
25. <article class='fill'>
26. <h3></h3>
27. + <img alt="Tipo de fuente" src="images/font.jpg" />
28. </article>
30. <article>
31. $ git merge origin/master
32. Updating 0486878..d0f88cc
33. Fast-forward
34. 2011-iim/sesion01/images/alineacion.jpg | Bin 0 -> 138057 bytes
35. 2011-iim/sesion01/images/espaciado.jpg | Bin 0 -> 86800 bytes
36. 2011-iim/sesion01/index.html | 6 +++---
37. 3 files changed, 3 insertions(+), 3 deletions(-)
38. create mode 100644 2011-iim/sesion01/images/alineacion.jpg
39. create mode 100644 2011-iim/sesion01/images/espaciado.jpg

**Examinar el historial de cambios de un archivo**

Podemos examinar todo el historial de cambios de un archivo mediante este comando:

1. git log --follow -p <archivo>

**Crear una nueva rama en el repositorio remoto**

1. git push origin origin:refs/heads/new-feature

**Diferencias entre 2 ramas de nuestro proyecto**

Para obtener las diferencias de 2 ramas de nuestro proyecto podemos utilizar este comando:

1. git diff master..experiment

**Manejar etiquetas (tags)**

Podemos gestionar etiquetas en nuestros proyectos de una manera muy sencilla, lo que nos ayudará a registrar de manera cómodo e intuitiva los puntos vitales de la evolución de nuestro proyecto.

Listar tags:

1. git tag

Crear un nuevo tag en el commit actual:

1. git tag -a v1.4 -m 'Versión 1.4'

Crear un nuevo tag en un commit pasado:

1. $ git tag -a v1.2 9fceb02

**Crear una rama local en el repositorio remoto:**

Es un comando un tanto extraño, si tenemos una rama local llamada "desarrollo" podemos enviarla al repositorio remoto origin

1. git push origin origin:refs/heads/desarrollo

**Averiguar quién hizo qué cosa en un archivo: git blame.**

1. $ git blame site.yaml
2. 9e34c40d (David Rubert 2011-12-09 09:25:04 +0100 1) handlers:
3. 7d8ccd71 (David Rubert 2011-12-22 19:43:37 +0100 2) - url: '/primeros-pasos/'
4. 7d8ccd71 (David Rubert 2011-12-22 19:43:37 +0100 3) static\_files: 'primeros-pasos/index.html'
5. 9e34c40d (David Rubert 2011-12-09 09:25:04 +0100 4) upload: '.\*'
6. 7d8ccd71 (David Rubert 2011-12-22 19:43:37 +0100 5) - url: '/trabajando-en-equipo/'
7. 7d8ccd71 (David Rubert 2011-12-22 19:43:37 +0100 6) static\_files: 'trabajando-en-equipo/index.html'
8. e7286e0d (David Rubert 2011-12-18 18:04:07 +0100 7) upload: '.\*'
9. e6d6195b (David Rubert 2012-01-08 22:12:17 +0100 8) - url: '/autenticacion-y-hosting/'
10. e6d6195b (David Rubert 2012-01-08 22:12:17 +0100 9) static\_files: 'autenticacion-y-hosting/index.html'
11. e6d6195b (David Rubert 2012-01-08 22:12:17 +0100 10) upload: '.\*'
12. 6f0dd636 (David Rubert 2012-01-12 23:13:47 +0100 11) - url: '/ejemplos/'
13. 6f0dd636 (David Rubert 2012-01-12 23:13:47 +0100 12) static\_files: 'ejemplos/index.html'
14. fde35e7f (David Rubert 2011-12-18 23:51:40 +0100 13) upload: '.\*'
15. 9e34c40d (David Rubert 2011-12-09 09:25:04 +0100 14) - url: '/'
16. 9e34c40d (David Rubert 2011-12-09 09:25:04 +0100 15) static\_files: 'index.html'
17. 9e34c40d (David Rubert 2011-12-09 09:25:04 +0100 16) upload: '.\*'
18. 9e34c40d (David Rubert 2011-12-09 09:25:04 +0100 17) - url: '/'
19. 9e34c40d (David Rubert 2011-12-09 09:25:04 +0100 18) static\_dir: '/'

**Servidor web con instaweb**

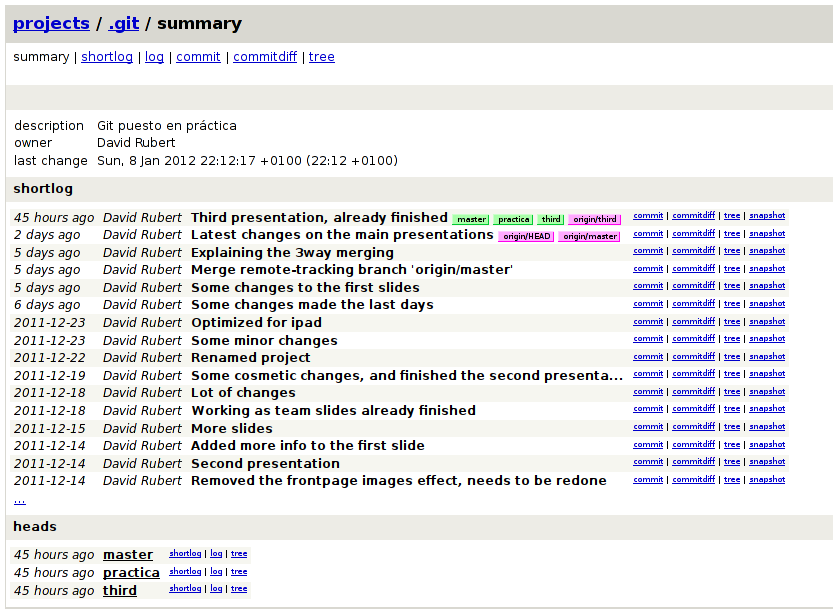
Podemos arrancar un servidor web que nos permita navegar el contenido de nuestro repositorio de una manera más amigable, para ello deberemos tener instalado **lighttpd** o cualquier otro servidor web compatible y ejecutar:

1. git instaweb

Para que no nos aparezca el mensaje de **Unnamed repository**:

1. echo "Mi proyecto" > .git/description

Veremos algo así:



Para pararlo:

1. git instaweb --stop