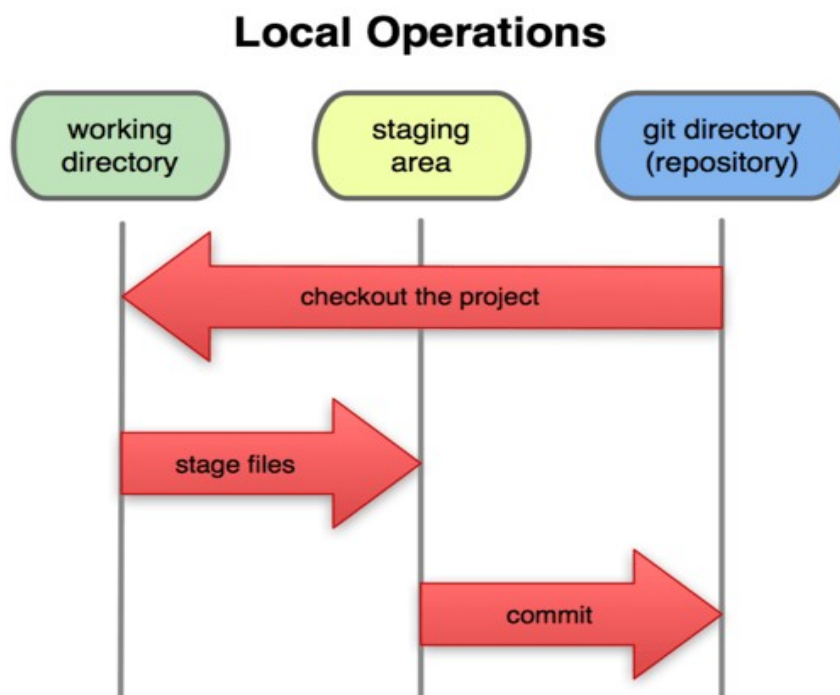


Versionado con Git en un repositorio local:

- 1) primer paso instalar git
- 2) verificar si se instalo:
`git version`
- 3) configurar nuestros datos Nombre de usuario y Correo electrónico:
`git config --global user.name "Christian Miranda"`
`git config --global user.email "chrisjav@hotmail.com"`
- 4) verificamos las configuraciones actuales luego de modificar:
`git config --list`
- 5) crear un directorio y abrir una terminal dentro
- 6) para que nuestro repositorio local sea administrado por git:
`git init`
- 7) tipear "`ls -alh`" para ver los cambios(incluyendo archivos y directorios ocultos) en el repositorio.
- 7) El comando "`git init`" creó un directorio con el historial de cambios
nota: si borramos ese directorio no git no podra administrarlo mas
- 8) Para definir una lista de archivos o directorios, que no queremos que se "pusheen" en el repositorio remoto con la posibilidad de ser accedidos por todos, tenemos que crear el archivo `.gitignore`
tipeamos `nano .gitignore` y escribimos dentro, lo que no queremos que se pushee al repositorio remoto (tambien se pueden usar wild cards, por ej: `*.txt`)
- 9) ver imagen "etapas Git.jpg"



Como podemos ver, nuestro repositorio local tiene 3 diferentes fases por la cual pasa nuestro código:

1. *Working Directory*
2. *Staging Area*
3. *Git Repository*

Al igual que las 3 fases, el código tiene 3 *estatus* en cada diferente fase:

1. **Modificado**
2. **Preparado**
3. **Confirmado**

Fase 1: “Working Directory”.

Aquí es donde podemos hacer cualquier cambio y no afecta a nuestro repositorio en lo absoluto. En cuanto modificamos algo de nuestro código, éste tiene status de **modificado**. Si ejecutamos el comando **git status**, nos mostrará qué archivos han sido modificados (o creados).

A terminal window with a dark background and light-colored text. The window title is 'christian@christian-Lenovo-V15-ADA: ~/git/mirepo'. The command 'git status' has been executed. The output shows the current branch is 'master', there are no commits yet, and there are files not being tracked, specifically '.gitignore'. It also provides instructions on how to use 'git add' to track these files.

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git status
En la rama master

No hay commits todavía

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
.gitignore

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa
"git add" para hacerles seguimiento)
christian@christian-Lenovo-V15-ADA:~/git/mirepo$
```

Una vez que hemos hecho los cambios necesarios, pasamos nuestros archivos al “**staging area**” con el comando:

git add -A // esto pasa al “staging area” todos los archivos “no trackeados”

o si escribimos: **git add .**

Agregamos todos los archivos modificados dentro de **working directory** a **staging area**.

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git add -A
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)
    nuevo archivo: .gitignore

christian@christian-Lenovo-V15-ADA:~/git/mirepo$
```

Cuando pasas el código de Working Directory a Staging Area, cambias el estado del código de **modificado** a **preparado**.

Para los casos en que no queremos que git siga trackeando determinado archivo del repositorio local debemos hacer por ejemplo:

git rm --cached .gitignore

Se puede ver como cambio el estado del archivo .gitignore:

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git rm --cached .gitignore
rm '.gitignore'
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git status
En la rama master

No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
    .gitignore

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa
"git add" para hacerles seguimiento)
christian@christian-Lenovo-V15-ADA:~/git/mirepo$
```

Lo volvemos a agregar con **git add .**

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git add .
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)
    nuevo archivo: .gitignore

christian@christian-Lenovo-V15-ADA:~/git/mirepo$
```

Fase 2: “Staging Area”

Aquí es donde le podemos dar nombre a nuestra nueva versión. Y crear una “copia” de cómo quedaría nuestro repositorio en producción.

Para pasar nuestro código de *staging area* al Git Repository (aun no se publica el código en Github), escribimos el siguiente comando:

git commit -m "Nombre del la nueva versión"

A terminal window with a dark background. The title bar shows 'christian@christian-Lenovo-V15-ADA: ~/git/mirepo'. The prompt is 'christian@christian-Lenovo-V15-ADA:~/git/mirepo\$'. The command 'git commit -m "primer commit"' has been entered. The output shows '[master (commit-raiz) f3a23fc] primer commit', '1 file changed, 1 insertion(+)', and 'create mode 100644 .gitignore'. The prompt is now 'christian@christian-Lenovo-V15-ADA:~/git/mirepo\$' with a cursor.

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git commit -m "primer commit"
[master (commit-raiz) f3a23fc] primer commit
1 file changed, 1 insertion(+)
create mode 100644 .gitignore
christian@christian-Lenovo-V15-ADA:~/git/mirepo$
```

Nota: cuando hacemos un commit el código pasa de estado **preparado** a **confirmado**.

Ahora escribimos el comando **git log**, el cual nos muestra un historial de todos los commits efectuados.

A terminal window with a dark background. The title bar shows 'christian@christian-Lenovo-V15-ADA: ~/git/mirepo'. The prompt is 'christian@christian-Lenovo-V15-ADA:~/git/mirepo\$'. The command 'git log' has been entered. The output shows 'commit f3a23fc7e2f9466f8c796ad1ffa7ff34f1b2c589 (HEAD -> master)', 'Author: Christian Miranda <licmirandachristian@gmail.com>', 'Date: Sun Feb 28 14:44:13 2021 -0300', and 'primer commit'. The prompt is now 'christian@christian-Lenovo-V15-ADA:~/git/mirepo\$' with a cursor.

```
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git log
commit f3a23fc7e2f9466f8c796ad1ffa7ff34f1b2c589 (HEAD -> master)
Author: Christian Miranda <licmirandachristian@gmail.com>
Date: Sun Feb 28 14:44:13 2021 -0300

    primer commit
christian@christian-Lenovo-V15-ADA:~/git/mirepo$
```

Fase 3. “Git repository”

Una vez que el código esta confirmado ya esta listo para sincronizarse con el servidor de Git (github, bitbucket,etc).

Para esto tenemos que crearnos un nuevo repositorio en github. Luego escribimos los comandos:

git branch -M main

git remote add origin https://github.com/chris33932/gitbasico.git

git push -u origin main

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo
christian@christian-Lenovo-V15-ADA:~/git/mirepo$ git push -u origin main
Username for 'https://github.com': chris33932
Password for 'https://chris33932@github.com':
Enumerando objetos: 10, listo.
Contando objetos: 100% (10/10), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (7/7), listo.
Escribiendo objetos: 100% (10/10), 1.07 KiB | 1.07 MiB/s, listo.
Total 10 (delta 1), reusado 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/chris33932/gitbasico.git
 * [new branch]      main -> main
Rama 'main' configurada para hacer seguimiento a la rama remota 'main' de 'origin'.
christian@christian-Lenovo-V15-ADA:~/git/mirepo$
```

Versionado con Git desde un repositorio Remoto:

Git clone, con este comando descargamos un repositorio que ya esta creado previamente.

Par ello nos situamos en un nuevo directorio, creado para tal fin, y escribimos:

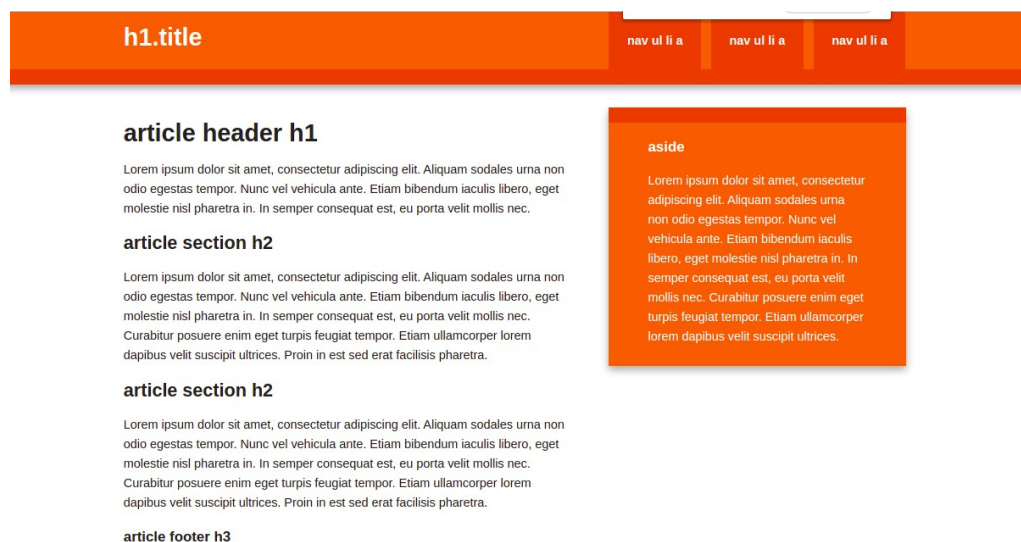
```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2
christian@christian-Lenovo-V15-ADA:~/git/mirepo2$ git clone https://github.com/c
hris33932/ejemplo.git
Clonando en 'ejemplo'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 27 (delta 1), reused 27 (delta 1), pack-reused 0
Desempaquetando objetos: 100% (27/27), 69.05 KiB | 166.00 KiB/s, listo.
christian@christian-Lenovo-V15-ADA:~/git/mirepo2$
```

como el repositorio es publico no pidió, loguearnos, luego hacemos **ls -l** para ver el nuevo directorio descargado, hacemos **cd ejemplo** para acceder al mismo luego hacemos un **git status**,


```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2$ ls -l
total 4
drwxrwxr-x 5 christian christian 4096 feb 28 16:58 ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2$ cd ejemplo/
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git status
En la rama master
Tu rama está actualizada con 'origin/master'.

nada para hacer commit, el árbol de trabajo está limpio
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

Ahora vamos a modificar el index.html dentro del repositorio;
actualmente es así:



Descarga: initializer.com

agregamos algunos cambios:



verificamos los cambios con git status

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git status
En la rama master
Tu rama está actualizada con 'origin/master'.

Cambios no rastreados para el commit:
(usa "git add <archivo>..." para actualizar lo que será confirmado)
(usa "git restore <archivo>..." para descartar los cambios en el directorio de trabajo)
    modificado:      css/main.css
    modificado:      index.html

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

el siguiente comando a utilizar es git diff, para ver que modificaciones ocurrieron en los archivos:

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git diff
diff --git a/css/main.css b/css/main.css
index 310f9e1..0e74680 100644
--- a/css/main.css
+++ b/css/main.css
@@ -81,7 +81,7 @@ body {
    .header-container,
    .footer-container,
    .main aside {
-       background: #f16529;
+       background: #29f1b2;
    }
}
```

se puede apreciar signos “-” “+”, que representan que se eliminó o que se agregó, hacemos lo siguiente para actualizar:

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git add .
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git commit -m 'modificando colores del index'
[master 59c4c9a] modificando colores del index
 2 files changed, 13 insertions(+), 13 deletions(-)
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git status
En la rama master
Tu rama está adelantada a 'origin/master' por 1 commit.
(usa "git push" para publicar tus commits locales)

nada para hacer commit, el árbol de trabajo está limpio
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git push
Username for 'https://github.com': chris33932
Password for 'https://chris33932@github.com':
Enumerando objetos: 9, listo.
Contando objetos: 100% (9/9), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (5/5), listo.
Escribiendo objetos: 100% (5/5), 642 bytes | 642.00 KiB/s, listo.
Total 5 (delta 3), reusado 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/chris33932/ejemplo.git
 60ebe5e..59c4c9a master -> master
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git status
En la rama master
Tu rama está actualizada con 'origin/master'.

nada para hacer commit, el árbol de trabajo está limpio
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

Utilizamos el comando **git push** para subir los cambios al repositorio remoto, en el caso de usar mas de un branch tenemos que agregar al comando lo siguiente: **git push origin master**.

Una buena práctica cuando trabajamos en un proyecto con mas de una persona, tenemos que hacer un **git pull**, para asegurarnos de que nuestro repositorio local se encuentra actualizado.

A terminal window with a dark background. The title bar shows 'christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo'. The prompt is 'christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo\$'. The user enters 'git pull'. The output is 'Ya está actualizado.' followed by a new prompt 'christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo\$'.

Creando una rama o “branch” en git, esto consiste en una bifurcación del estado del código, es decir se crea un nuevo camino para el mismo. Con Git Branch es posible crear diferentes ramas de desarrollo que pueden converger en el mismo repositorio.

Para crear una branch hacemos:

A terminal window with a dark background. The title bar shows 'christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo'. The prompt is 'christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo\$'. The user enters 'git branch minuevarama'. The output is 'christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo\$' followed by a new prompt 'christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo\$'.

luego nos pasamos a la nueva rama creada:

A terminal window with a dark background. The title bar shows 'christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo'. The prompt is 'christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo\$'. The user enters 'git branch minuevarama' and then 'git checkout minuevarama'. The output is 'Cambiado a rama 'minuevarama'' followed by a new prompt 'christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo\$'.

hacemos git status :


```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git branch minuevarama
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git checkout minuevarama
Cambiado a rama 'minuevarama'
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git status
En la rama minuevarama
nada para hacer commit, el árbol de trabajo está limpio
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

escribimos **git branch** para ver todas las branch que existen, en nuestra maquina, y para ver además las que existen de manera remota agregamos **-a** al final.

Luego modificamos el código nuevamente, y nos queda:



actualizamos git pero esta vez desde la nueva rama:

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git status
En la rama minuevarama
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de trabajo)
    modificado:    css/main.css

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git add -A
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git commit -m 'modificando el color del index'
[minuevarama 543c763] modificando el color del index
1 file changed, 1 insertion(+), 1 deletion(-)
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

luego escribimos el comando **git push -u origin minuevarama**, para avisar a nuestra

copia local de git que existe una nueva rama en el repositorio remoto que va a estar asociada(linkeada) con mi branch en repositorio local

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git push -u origin minuevarama
Username for 'https://github.com': chris33932
Password for 'https://chris33932@github.com':
Enumerando objetos: 7, listo.
Contando objetos: 100% (7/7), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (4/4), listo.
Escribiendo objetos: 100% (4/4), 424 bytes | 424.00 KiB/s, listo.
Total 4 (delta 2), reusado 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'minuevarama' on GitHub by visiting:
remote:   https://github.com/chris33932/ejemplo/pull/new/minuevarama
remote:
To https://github.com/chris33932/ejemplo.git
 * [new branch]      minuevarama -> minuevarama
Rama 'minuevarama' configurada para hacer seguimiento a la rama remota 'minuevarama' de 'origin'.
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

hacemos un git log:

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git log
commit 543c763974d871a55b88afe3339f8efdb8d310bd (HEAD -> minuevarama, origin/minuevarama)
Author: Christian Miranda <licmirandachristian@gmail.com>
Date:   Sun Feb 28 17:54:53 2021 -0300

    modificando el color del index

commit 59c4c9afb3f8dd0ede87e1e0876aa7d833450889 (origin/master, origin/HEAD, master)
Author: Christian Miranda <licmirandachristian@gmail.com>
Date:   Sun Feb 28 17:33:59 2021 -0300

    modificando colores del index

commit 60ebe5e309f1f8416714eaf3e710d27ba27fd717
Author: Christian Miranda <licmirandachristian@gmail.com>
Date:   Sun Feb 28 16:51:06 2021 -0300

    primer commit
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

Nota: todavía seguimos en nuestra nueva rama, esto quiere decir que si otra persona se baja el código del repositorio remoto no tendrá acceso a nuestras modificaciones.

Si hacemos **git branch -a**, vemos las branch locales y remotas

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git branch -a
master
* minuevarama
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/minuevarama
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

Lo siguiente que vamos a hacer es unir nuestra nueva rama a la master para que los demás integrantes del proyecto puedan acceder a nuestra modificación.

Para lo cual nos volvemos a pasar a la rama master, luego hacemos un origin pull a la master para asegurarnos que contamos con la última versión del código:

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git checkout master
Cambiado a rama 'master'
Tu rama está actualizada con 'origin/master'.
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git pull origin master
Desde https://github.com/chris33932/ejemplo
* branch      master      -> FETCH_HEAD
Ya está actualizado.
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

Luego hacemos **git branch --merged**, para ver cuáles son las branch que hemos “mergeado” (fusionado).

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git branch --merged
* master
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

se puede ver que solo aparece la master, ya que todavía no hemos “mergeado” la branch que acabamos de crear. Posteriormente pasamos a “mergear”:

con el comando **git merge minuevarama**, podemos ver como unió los cambios a la rama master.

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git branch --merged
* master
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git merge minuevarama
Actualizando 59c4c9a..543c763
Fast-forward
 css/main.css | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

Una vez que hicimos esto debemos subir el código mergeado al servidor o repositorio remoto, con el comando:

git push origin master

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git push origin master
Username for 'https://github.com': chris33932
Password for 'https://chris33932@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/chris33932/ejemplo.git
  59c4c9a..543c763  master -> master
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

lo siguiente que vamos a hacer será borrar la branch que hemos creado, ya que no la vamos a volver a usar para lo cual escribimos los comandos:

git push origin --delete minuevarama

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git push origin --delete minuevarama
Username for 'https://github.com': chris33932
Password for 'https://chris33932@github.com':
To https://github.com/chris33932/ejemplo.git
- [deleted]          minuevarama
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```

acabamos de borrar la nueva branch del repositorio remoto, lo siguiente que haremos será borrarla del repositorio local con los siguientes comandos(solo si lo consideran necesario, ya que almacenar ramas en modo local no genera inconvenientes para la administración por parte de Git):

```
christian@christian-Lenovo-V15-ADA: ~/git/mirepo2/ejemplo
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$ git branch -d minuevarama
Eliminada la rama minuevarama (era 543c763)..
christian@christian-Lenovo-V15-ADA:~/git/mirepo2/ejemplo$
```