Comandos Git

**git init:** para iniciar repositorio en la carpeta seleccionada, se pone al principio. Cuando comenzamos el proyecto git crea dos áreas un área de ensayo (temporal nos sirve para conocer el estado de los archivos) y una repositorio local que es donde se almacenan las instantáneas para ser rescatadas cuando queramos.

**git status -s:** nos da un listado con todos los archivos en la carpeta del proyecto que están o no en seguimiento MUY IMPORTANTE para consultar.

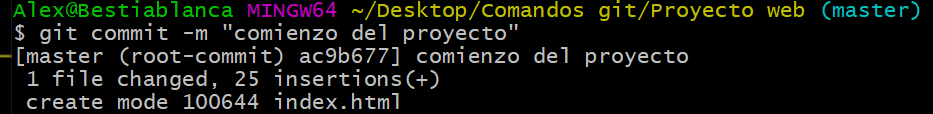


**git add (nombre del archivo):** nos hace el seguimiento del archivo seleccionado. Se puede ver en la imagen como agregamos el archivo en concreto en este caso el index.html y luego al comprobar el estado con status -s nos aparece una A de color verde indicando que ese archivo ha sido agregado (ahora está en el área de ensayo)

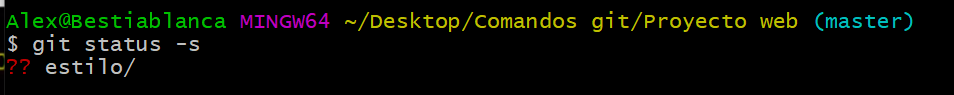


**git add .** (git add “ESPACIO” PUNTO) a todos los archivos conseguimos que lleve el archivo desde el directorio de trabajo a el área de ensayo.

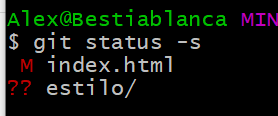
**git commit -m “descripción”:** foto fija del proyecto una vez la tenemos en seguimiento normalmente ponemos el estado de la copia que queremos guardar en este caso el comienzo del proyecto index.html



Si volvemos a hacer un git status -s, ya no nos aparece el index.html porque sí que se le está haciendo un seguimiento

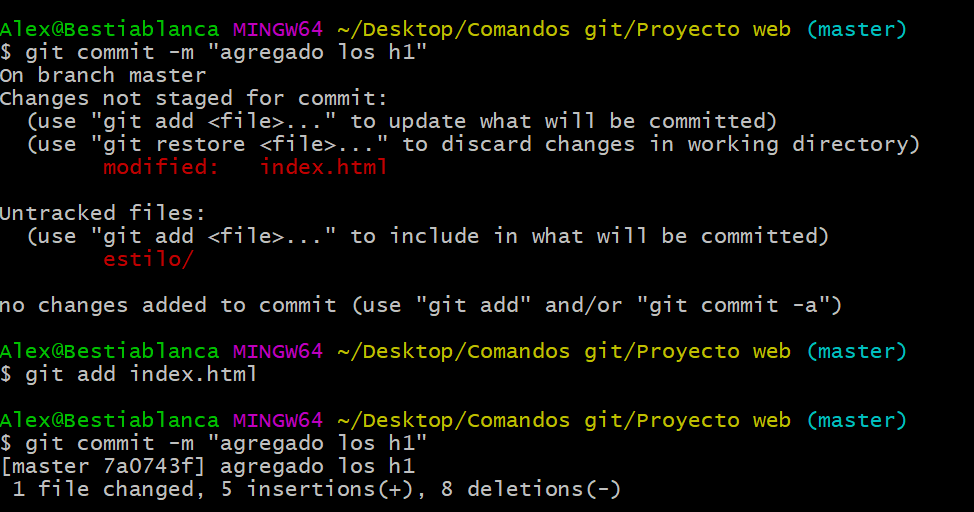


Si modificamos el archivo y volvemos a hacer git status -s, nos aparece que sí que ha sido modificado

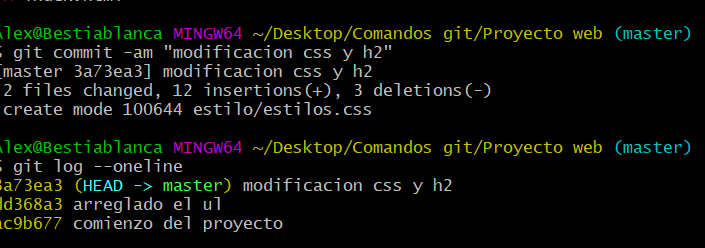


Se puede observar como da error si directamente queremos guardar la modificación del archivo previamente debemos de indicar a git que queremos añadir el archivo index.html con el comando git add, una vez el archivo ha sido añadido ya podemos entonces agregar la captura de pantalla con el

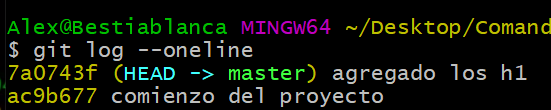
git commit-m



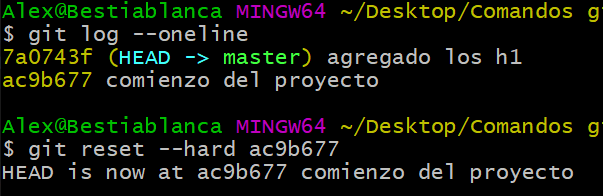
**git commit -am** **“descripcion”:** hacerlo a la vez tanto el add como el -m de esta manera en vez de ir uno por uno y paso a paso lo hacemos al a vez es la misma operación que añadir previamente con git add nombre del archivo o git add . y luego hacer el git commit -m, pero a la vez.



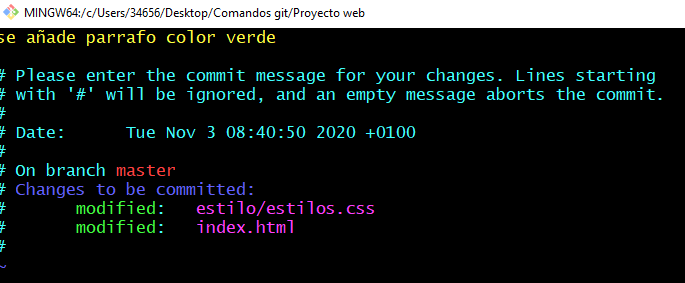
**git log - -oneline**: nos aparecerán las fotos fijas. Nos aparecen las copias hechas con el commit -m sirve de consulta



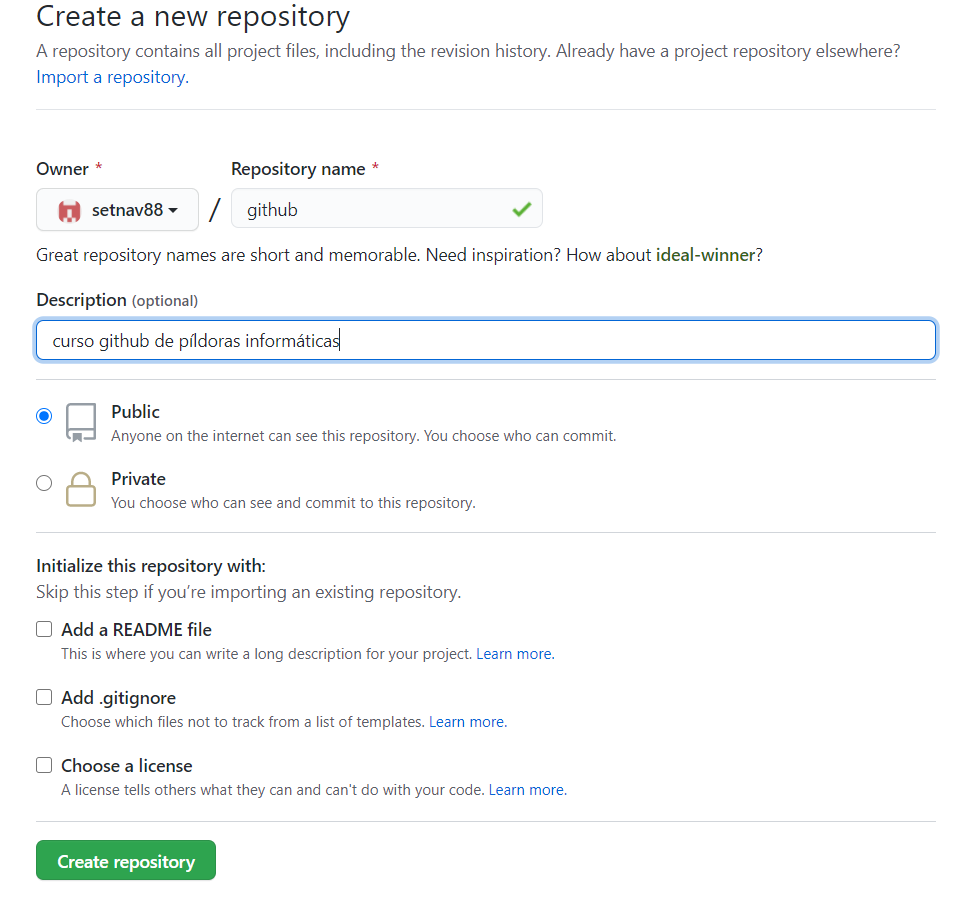
**git reset - -hard seleccionar código instantanea** : volver a la foto fija/versión anterior. De esta manera como se puede observar seleccionamos con git reset –hard y el código la primera captura guardada del proyecto



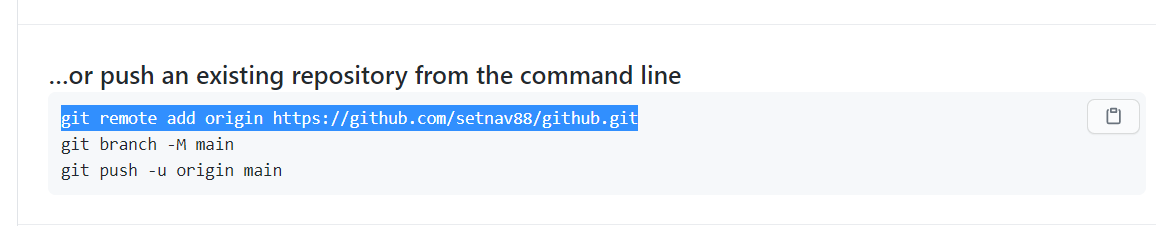
**git commit –-(dos guiones) amend**: nos abre el editor bin para modificar, debemos de poner :i para que nos permita modificar luego cambiamos el texto apretamos escape y listo para que quede guardado :wq



Ahora accedemos a [www.github.com](http://www.github.com) y pinchamos en crear repositorio. Ponemos el nombre que queremos que tenga el repositorio y una pequeña descripción el nombre irá previamente con nuestro nombre de usuario.

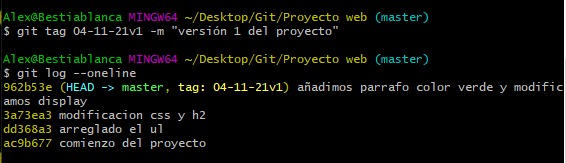


Al tratarse de un repositorio nuestro debemos de copiar ese enlace en git para que lo suba si en la consola no nos aparece que nos identifiquemos deberemos de introducir $ git push origin master de esta manera forzamos a que nos pida el usuario y contraeña.

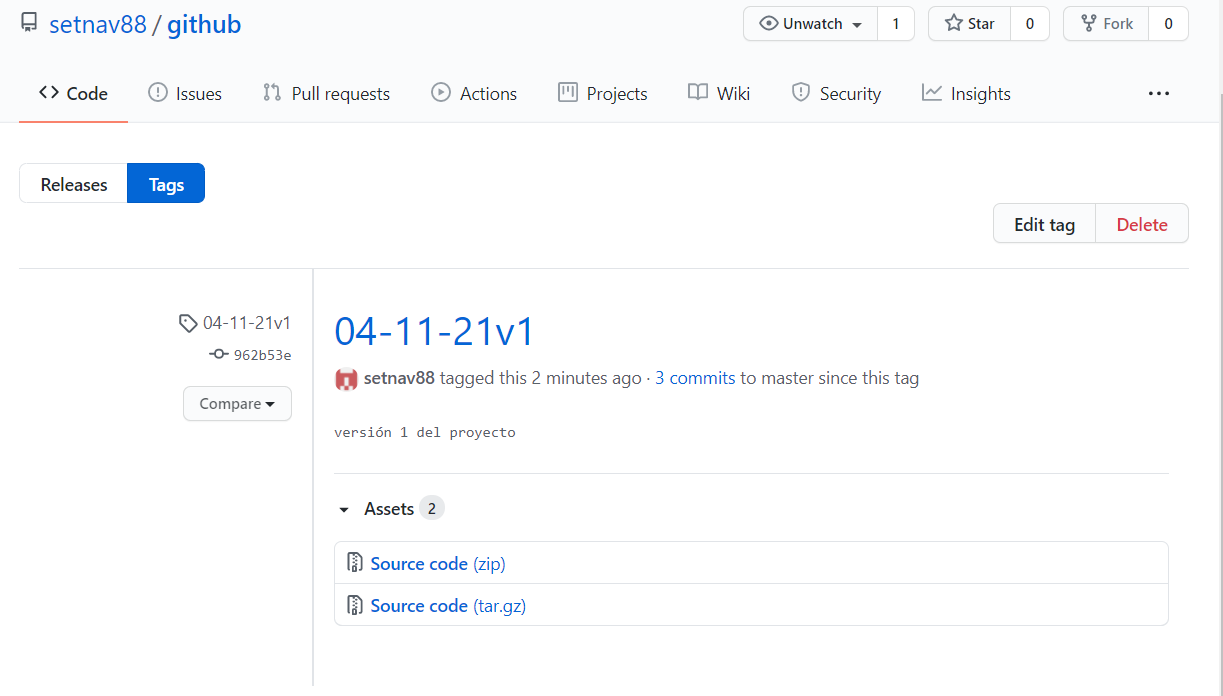




**git tag 05-10-20v1 -m "Versión 1 del proyecto":** para guardar un tag en este caso la primera versión



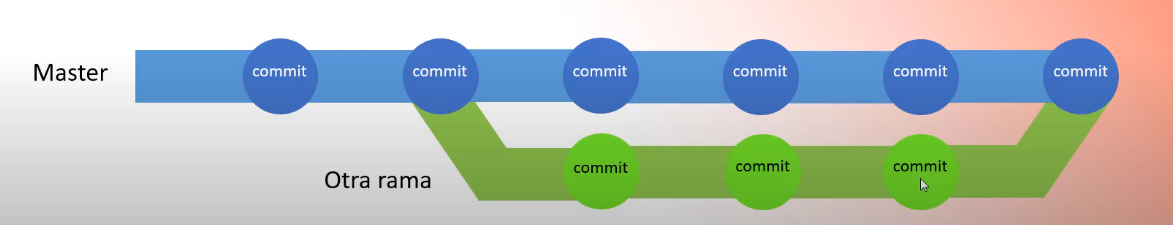
**git tag push:** Para subir las tag al repositorio remoto.



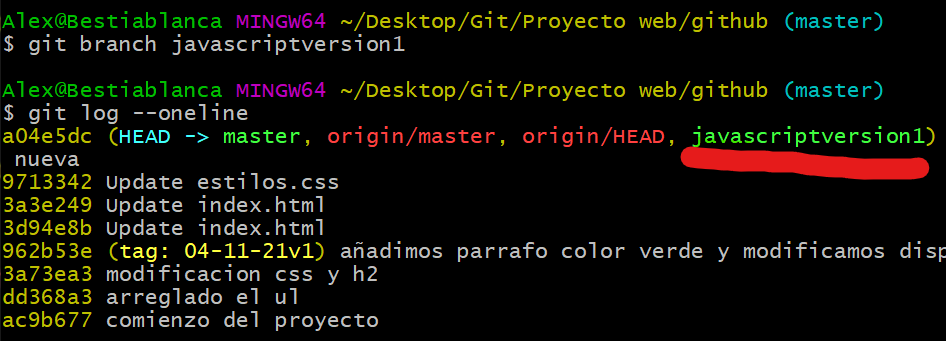
**git clone** y url copiada desde la web de github para clonar poyecto.

A las ramas o líneas de tiempo se les llama Branch:

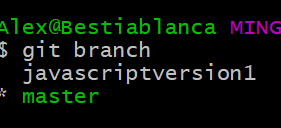
Master es el nombre a la línea de tiempo o rama que git. La rama Master es la que crea git por defecto, pero podemos crear más ramas a parte de la principal



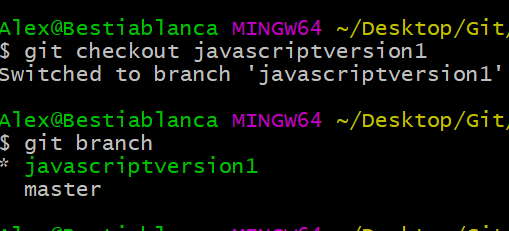
Para crear una rama únicamente le damos a git Branch y le ponemos el nombre que queramos. Se puede ver como tenemos nuestra rama creada cuando hacemos un git log –oneline



Podemos ver en qué rama estoy trabajando con “git Branch” se ve de color verde y con \*



Si queremos acceder al nuevo Branch o rama debemos de escribir git checkout y el nombre de la rama en nuestro caso git checkout javascriptversion1, luego hacemos comprobación nuevamente con git Branch y ya nos aparece seleccionado en verde la nueva línea con el \* correspondiente



Si queremos juntar tanto la rama master como la rama Branch pasamos a la rama master git checkout master y una vez dentro, utilizamos el merge con el nombre del Branch secundario en este caso

git merge javascriptversion1

