

Laboratorio 1.

1. Flujo de trabajo con permiso de escritura en el repositorio compartido.

Lo primero que hacemos en esta práctica es clonar el repositorio compartido usando el comando **git clone** seguido de la url del repositorio:

```
(base) pablobv@f-l3209-pc30:~/ISI/lab_1/practica-isi$ git clone https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi.git
Clonando en 'practica-isi'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (6/6), 546 bytes | 11.00 KiB/s, listo.
```

Una vez que tenemos el repositorio en local, creamos una nueva rama, en mi caso esa rama se llama **pablobv/nuevos.requisitos**. Para crear dicha rama hacemos uso del comando **git branch** junto con el nombre de la rama, posteriormente, nos cambiaremos a dicha rama aplicando el comando **git checkout pablobv/nuevos.requisitos**:

```
(base) pablobv@f-l3202-pc08:~/ISI/lab_1/practica-isi$ git branch
* master
  pablobv/nuevos.requisitos
  revisar-cambios
(base) pablobv@f-l3202-pc08:~/ISI/lab_1/practica-isi$ git checkout pablobv/nuevos.requisitos
Cambiado a rama 'pablobv/nuevos.requisitos'
```

Tras esto, ya estaremos en la rama que hemos creado y es en ella donde vamos a efectuar alguna modificación al proyecto en el que estamos trabajando. Una vez que hayamos realizado las modificaciones, lo que hacemos es añadir dichos cambios y subirlos al repositorio en remoto (**ATENCIÓN: seguimos estando en la rama pablobv/nuevos.requisitos**, así que **estos cambios SÓLO** deben aparecer en mi rama, **NO** esarán en el main).

Los comandos usados para esta parte son:

1º **git add --all**

2º **git commit -am "Añadiendo a Pablo a members"**

3º **git push origin pablobv/nuevos.requisitos**

```
(base) pablobv@f-l3202-pc08:~/ISI/lab_1/practica-isi$ git add --all
```

```
Terminal
(base) pablobv@f-l3202-pc08:~/ISI/lab_1/practica-isi$ git branch
master
* pablobv/nuevos.requisitos
revisar-cambios
(base) pablobv@f-l3202-pc08:~/ISI/lab_1/practica-isi$ git commit -am "Añadiendo a Pablo a members"
[pablobv/nuevos.requisitos 069a535] Añadiendo a Pablo a members
1 file changed, 1 insertion(+)
(base) pablobv@f-l3202-pc08:~/ISI/lab_1/practica-isi$ git push origin pablobv/nuevos.requisitos
Username for 'https://gitlab.etsit.urjc.es': pablobv
Password for 'https://pablobv@gitlab.etsit.urjc.es':
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 6 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 381 bytes | 190.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
remote:
remote: To create a merge request for pablobv/nuevos.requisitos, visit:
remote: https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi/-/merge_requests/new?merge_request%5Bsource_branch%5D=pablobv%2Fnuevos.requisitos
remote:
```

Una vez hecho esto, lo que hay que hacer es hablar con un compañero para que haga un pull en su copia del repositorio compartido, con el objetivo de que la mezcle en la rama principal y suba la rama principal al repositorio compartido. Esta metodología de trabajo es bastante común, ya que así el otro usuario **puede supervisar en su copia local los cambios que yo he hecho y, en caso de que cuente con su visto bueno, subirla al repositorio compartido.** En cuanto a los pasos que mi compañero debería seguir, estos son los siguientes:

1º Git remote add brosaa/origin

2º Git fetch origin brosaa/nuevos-requisitos (NOTA: la rama es la de la compañera con la que se realizó la simulación)

3º Git branch revisar-cambios origin/brosaa/nuevos-requisitos

4º Git checkout revisar-cambios

5º Abrir el archivo(cat members.txt) para comprobar si se han aplicado los cambios.

6º Git checkout master

7º Git merge revisar-cambios

8º Git push

```
(base) pablobv@f-l3209-pc30:~/ISI/lab_1/practica-isi$ git remote add brosaa origin
```

```
(base) pablobv@f-l3209-pc30:~/ISI/lab_1/practica-isi$ git fetch origin brosaa/nuevos-requisitos
Desde https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi
* branch brosaa/nuevos-requisitos -> FETCH_HEAD
```

```
(base) pablobv@f-l3209-pc30:~/ISI/lab_1/practica-isi$ git branch revisar-cambios origin/brosaa/nuevos-requisitos
Rama 'revisar-cambios' configurada para hacer seguimiento a la rama remota 'brosaa/nuevos-requisitos' de 'origin'.
```

```
(base) pablobv@f-l3209-pc30:~/ISI/Lab_1/practica-isi$ git checkout revisar-cambios
Cambiado a rama 'revisar-cambios'
Tu rama está actualizada con 'origin/brosea/nuevos-requisitos'.
(base) pablobv@f-l3209-pc30:~/ISI/Lab_1/practica-isi$ cat members.txt
Juan Antonio Ortega Aparicio
Belén Rosa Alonso
(base) pablobv@f-l3209-pc30:~/ISI/Lab_1/practica-isi$ git checkout master
Cambiado a rama 'master'
Tu rama está actualizada con 'origin/master'.
(base) pablobv@f-l3209-pc30:~/ISI/Lab_1/practica-isi$ git merge revisar-cambios
Actualizando a3ed40d..87c02f7
Fast-forward
 members.txt | 1 +
 1 file changed, 1 insertion(+)
```

Estos comandos lo que hacen básicamente es bajarse la rama de la otra persona (1 y 2), crear una rama que será donde revisemos los cambios de la otra persona y las enlazamos(3), nos cambiamos a dicha rama(4) y comprobamos los cambios(5), en este caso como simplemente es añadir una línea de texto lo que haremos es volver a la rama **master(6)** y en ella unir la rama revisar-cambios a la master(7) para posteriormente subirlo(8).

2. Flujo de trabajo sin permiso de escritura en el repositorio compartido.

La principal diferencia que tiene esta parte con respecto a la anterior es que **no tenemos permiso de escritura, ya que las modificaciones que se van a hacer son sobre el proyecto de otro grupo**. Por ello, se hace uso de **merge requests**, que son peticiones a miembros de otro grupo para que acepten los cambios que has implementado en el mismo.

En primer lugar, dado que no tendremos de forma local el proyecto del otro grupo, lo que hay que hacer es un **fork**, que no es más que un duplicado del repositorio en el cual puedo hacer cambios sin que estos afecten al proyecto original. Para ello, situándonos en el repositorio que queremos clonar, le damos a fork.

Tras haber hecho la copia, lo que hacemos es, como siempre, clonar nuestro repositorio para acceder a su contenido en local.

Una vez hecho este proceso básico, creamos una nueva rama en la que vamos a hacer alguna modificación con respecto a lo que tenía el otro grupo en su repositorio, luego hacemos un commit de esos cambios y los subimos a nuestro proyecto forkeado haciendo un push. Comandos:

1º git branch pablobv/nuevos-requisitos (NOTA: pablobv/nuevos requisitos es el nombre de la rama)

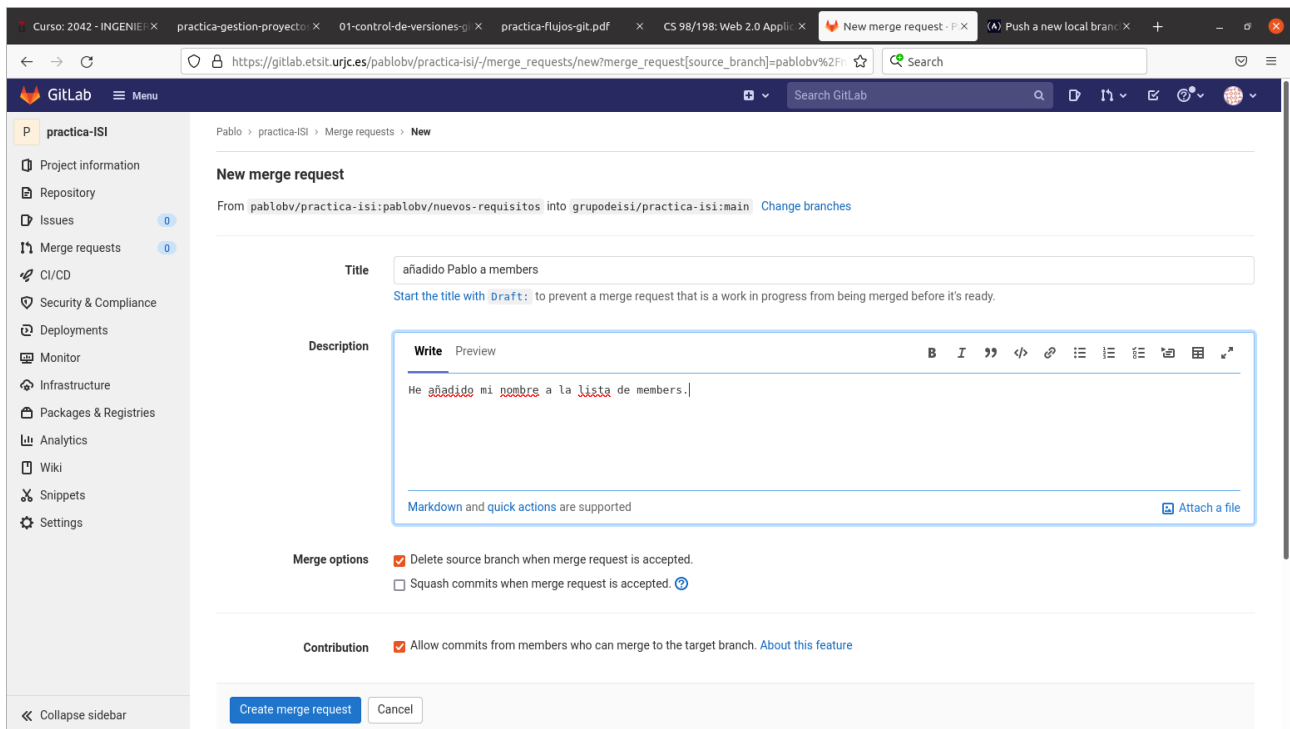
2º git checkout pablobv/nuevos-requisitos

3º git commit -am "añadido Pablo a members"

```
(base) pablobv@f-l3209-pc37:~/ISI/practica-isi$ git branch
* main
(base) pablobv@f-l3209-pc37:~/ISI/practica-isi$ git branch pablobv/nuevos-requisitos
(base) pablobv@f-l3209-pc37:~/ISI/practica-isi$ git branch
* main
  pablobv/nuevos-requisitos
(base) pablobv@f-l3209-pc37:~/ISI/practica-isi$ git checkout pablobv/nuevos-requisitos
Cambiado a rama 'pablobv/nuevos-requisitos'
(base) pablobv@f-l3209-pc37:~/ISI/practica-isi$ cat memebbers.txt
Samuel Martín Ortega
Paula Pulido Praena
Amelia Sánchez Aparicio

Belén Rosa Alonso - Grupo externo, curso 2017
César Borao Moratinos - Grupo externo
Juan Antonio Ortega Aparicio - Grupo externo
Pablo Barquero Villanueva- Grupo externo
(base) pablobv@f-l3209-pc37:~/ISI/practica-isi$ git commit -am "añadido Pablo a members"
[pablobv/nuevos-requisitos 67d618b] añadido Pablo a members
```

Una vez que ya tenemos los cambios en nuestro repositorio, como se ha comentado antes, vamos a hacer un **merge request** para que nos los acepten:




(Nota: cuando subimos los cambios sale un mensaje para crear el merge request, ya que detecta que proviene de otro repositorio en el que no tengo permiso de escritura).

En este caso, como se nos solicita que rechazemos inicialmente el merge request y pongamos un comentario, eso es lo que hacemos llendo a la pestaña de merge requests y viendo que tenemos uno en estado **open**, es decir, no se ha aceptado ni rechazado.




Tras poner el comentario y ver que nos rechazan el merge request nos ponemos a hacer los cambios que nos piden en el comentario y repetimos los pasos anteriores, viéndose en el apartado **Changes** dentro del merge request algo similar a esto:




Open Created 4 minutes ago by  **PULIDO PRAENA, PAULA** Contributor



Edit **Mark as draft** 

nueva prueba de merge

Overview **1** Commits **2** **Changes 1**

 Compare master  and latest version 

 **members.txt**  **+2 -1** ☐ Viewed 

 Show all unchanged lines  Show 20 lines

2	2	Belén Rosa Alonso
3	3	Pablo Barquero Villanueva - Belén acepta
4	4	
5	-	
6	5	hola a todos
	6	+
	7	+ vuelvo a modificar

Nota: aquí lo que debemos observar es que el comentario que hizo el miembro del grupo fue que borrarase una línea y que pusiese vuelvo a modificar.

Posteriormente, el usuario que quiere aceptar el merge request debe bajarse a su proyecto clonado la rama del que solicita el merge request y aplicar los siguientes comandos:

1º git fetch “[git@gitlab.etsit.urjc.es:ppulido/p.git](https://gitlab.etsit.urjc.es/ppulido/p.git)” ‘ppulido/nuevos-requisitos’

2º git checkout -b ‘ppulido/p-ppulido/nuevos-requisitos FETCH_HEAD

3º git branch

4º git fetch origin

5º git checkout ‘master’

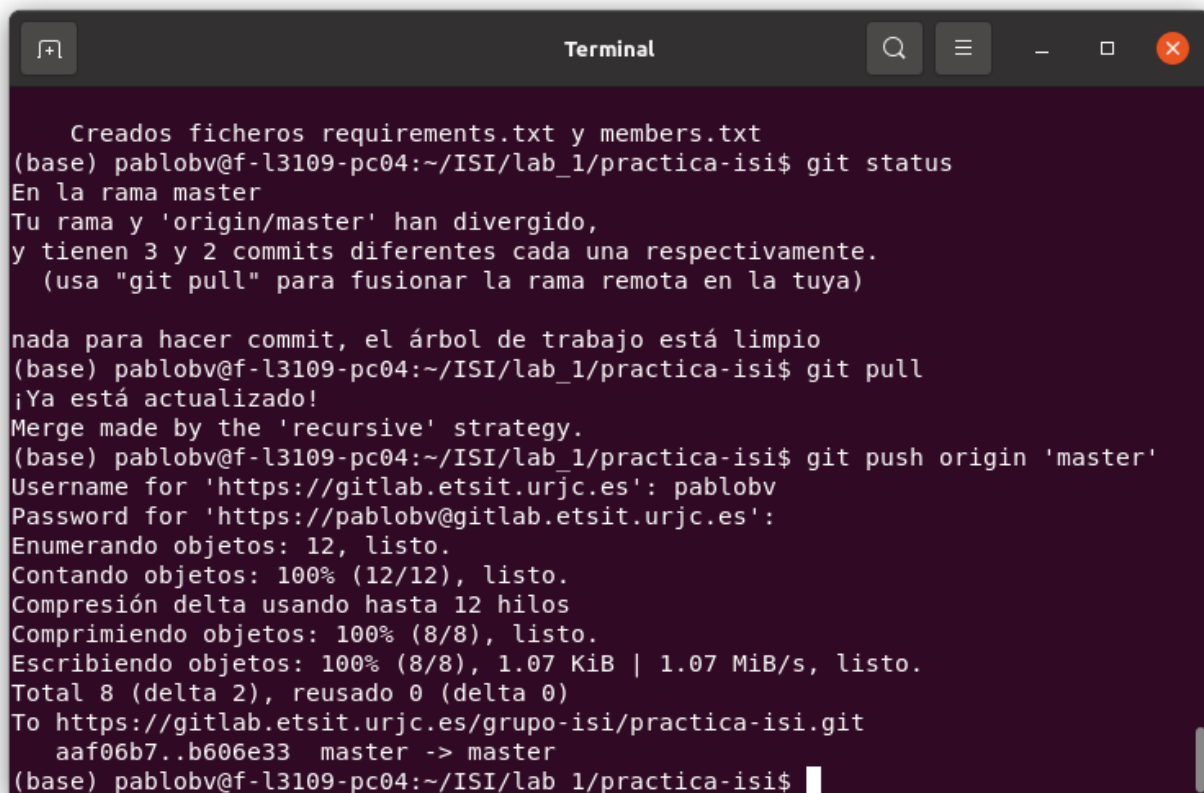
6º git merge -no-ff ‘ppulido/p-ppulido/nuevos-requisitos’

7º git push origin ‘master’

```
(base) pablov@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git fetch "git@gitlab.etsit.urjc.es:ppulido/p.git" 'ppulido/nuevos-requisitos'
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 9 (delta 3), reused 0 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (9/9), 919 bytes | 15.00 KiB/s, listo.
Desde gitlab.etsit.urjc.es:ppulido/p
* branch                ppulido/nuevos-requisitos -> FETCH_HEAD
(base) pablov@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git checkout -b 'ppulido/p-ppulido/nuevos-requisitos' FETCH_HEAD
Cambiado a nueva rama 'ppulido/p-ppulido/nuevos-requisitos'
(base) pablov@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git branch
master
* ppulido/p-ppulido/nuevos-requisitos
```

```
(base) pablobv@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git fetch origin
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 1 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (2/2), 527 bytes | 47.00 KiB/s, listo.
Desde https://gitlab.etsit.urjc.es/grupo-isi/practica-isi
0fad43c..aaf06b7 master -> origin/master
```

```
(base) pablobv@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git checkout 'master'
Cambiado a rama 'master'
Tu rama está detrás de 'origin/master' por 3 commits, y puede ser avanzada rápidamente.
(usa "git pull" para actualizar tu rama local)
(base) pablobv@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git merge --no-ff 'ppulido/p-pulido/nuevos-requisitos'
Merge made by the 'recursive' strategy.
 members.txt | 4 +++++
 1 file changed, 4 insertions(+)
```



```

Creados ficheros requirements.txt y members.txt
(base) pablobv@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git status
En la rama master
Tu rama y 'origin/master' han divergido,
y tienen 3 y 2 commits diferentes cada una respectivamente.
(usa "git pull" para fusionar la rama remota en la tuya)

nada para hacer commit, el árbol de trabajo está limpio
(base) pablobv@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git pull
¡Ya está actualizado!
Merge made by the 'recursive' strategy.
(base) pablobv@f-l3109-pc04:~/ISI/lab_1/practica-isi$ git push origin 'master'
Username for 'https://gitlab.etsit.urjc.es': pablobv
Password for 'https://pablobv@gitlab.etsit.urjc.es':
Enumerando objetos: 12, listo.
Contando objetos: 100% (12/12), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (8/8), listo.
Escribiendo objetos: 100% (8/8), 1.07 KiB | 1.07 MiB/s, listo.
Total 8 (delta 2), reusado 0 (delta 0)
To https://gitlab.etsit.urjc.es/grupo-isi/practica-isi.git
aaf06b7..b606e33 master -> master
(base) pablobv@f-l3109-pc04:~/ISI/lab_1/practica-isi$
```

Nota: estos comandos se pueden encontrar en el merge request, el usuario que quiere aceptarlo tiene una pestaña en la que aparecen todos estos comandos.

Por último, ahora el usuario que ha solicitado el merge request podrá ver que sus cambios ya han sido integrados, así que si realiza un pull de la rama principal del repositorio compartido podrá ver que aparecen ya dichos cambios. Si tras esto hacemos un push a nuestro proyecto forkeado, podremos ver que

tanto en dicho proyecto, como en el local tendremos el mismo contenido que en el repositorio compartido.