

# Flujos de trabajo en equipo con git

## Laboratorio 1

Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación, URJC

### Tabla de contenidos

1. Flujo de trabajo con permiso de escritura en el repositorio compartido .....	1
1.1. Sube cambios que deben ser aceptados por otro compañero de tu equipo.....	1
1.2. Acepta cambios de un compañero de tu equipo .....	3
2. Flujo de trabajo sin permiso de escritura en el repositorio compartido .....	4
2.1. Sube cambios que deben ser aceptados por otro compañero de otro equipo .....	4
2.2. Acepta cambios de un compañero de otro equipo.....	8

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

### 1.1. Sube cambios que deben ser aceptados por otro compañero de tu equipo

En este apartado, vamos a trabajar en el repositorio GitLab de nuestro equipo al cual tiene acceso el profesor y tiene la siguiente URL: <https://gitlab.etsit.urjc.es/grupo-isi/practica-isi>. El objetivo es realizar cambios en los archivos del repositorio y que un compañero del equipo me acepte dichos cambios, a pesar de que yo puedo incluir cambios directamente.

1. Clonamos el repositorio compartido del equipo en nuestro entorno local con el comando *git clone url\_repositorio* y tendremos un directorio idéntico al repositorio de GitLab en nuestro equipo local.

```
(base) brosaaf-l3209-pc29:~$ git clone https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi.git
Clonando en 'practica-isi'...
Username for 'https://gitlab.etsit.urjc.es': brosaaf-l3209-pc29
Password for 'https://brosaaf-l3209-pc29@gitlab.etsit.urjc.es':
```

Ilustración 1: Clonado del repositorio grupal en terminal

2. Entramos en el repositorio clonado y creamos una nueva rama local mediante el comando *git branch brosaaf/nuevos-requisitos* y entramos en ella mediante el comando *git checkout brosaaf/nuevos-requisitos*. Con el comando *git branch* podemos comprobar las ramas existentes, donde veremos en color verde y un asterisco (\*) la rama donde estamos situados.

```
(base) brosaaf-l3209-pc29:~/practica-isi$ git branch brosaaf/nuevos-requisitos
(base) brosaaf-l3209-pc29:~/practica-isi$ git checkout brosaaf/nuevos-requisitos
Cambiado a rama 'brosaaf/nuevos-requisitos'
(base) brosaaf-l3209-pc29:~/practica-isi$ git branch
* brosaaf/nuevos-requisitos
master
```

Ilustración 2: Creación de nueva rama local

3. Modificamos el archivo `members.txt` añadiendo una línea con mi nombre. A continuación, hago `git add` y añado un commit con `git commit -m`.

```
(base) brosaaf-l3209-pc29:~/practica-isi$ git add --all
(base) brosaaf-l3209-pc29:~/practica-isi$ git commit -m "Belen añadida a members"
[brosoa/nuevos-requisitos 87c02f7] Belen añadida a members
1 file changed, 1 insertion(+)
```

Ilustración 3: Commit de la modificación de archivos en terminal

4. Hago `git push` de mi rama local hacia el repositorio grupal de GitLab y se crea el commit en la URL <https://gitlab.etsit.urjc.es/grupo-isi/practica-isi/-/commit/87c02f7db8de353df69279da82bc7f0f153c6c3b>.

```
(base) brosaaf-l3209-pc29:~/practica-isi$ git push origin brosoa/nuevos-requisitos
Username for 'https://gitlab.etsit.urjc.es': brosoa
Password for 'https://brosoa@gitlab.etsit.urjc.es':
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 4 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 340 bytes | 340.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
remote:
remote: To create a merge request for brosoa/nuevos-requisitos, visit:
remote: https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi/-/merge_requests/new?merge_request%5Bsource_branch%5D=brosoa%2Fnuevos-requisitos
remote:
To https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi.git
* [new branch] brosoa/nuevos-requisitos -> brosoa/nuevos-requisitos
```

Ilustración 4: Git push de la nueva rama a GitLab en terminal

Al no existir esta rama en el repositorio, se crea una nueva con el nombre y podemos acceder a ella a través de GitLab (ilustración 5, cuadros verdes). Al hacer push de este modo, no mezclamos automáticamente nuestras modificaciones en la rama `main` (o `master`) del repositorio, ésta rama seguirá igual y las modificaciones estarán en la rama `brosoa/nuevos-requisitos`.

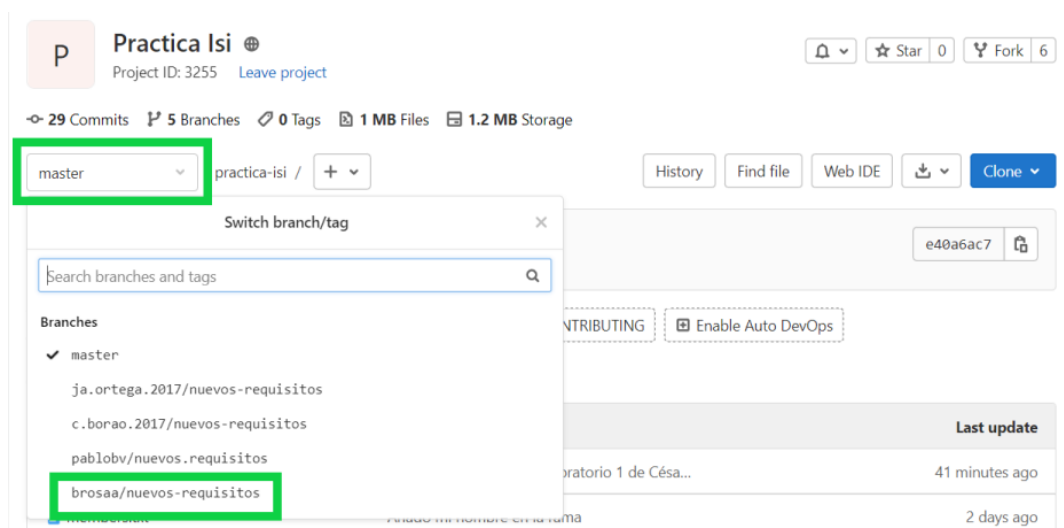


Ilustración 5: Ramas en el repositorio GitLab

5. A continuación, un compañero del equipo habrá hecho pull de nuestra rama en su repositorio clonado local y tras revisar mis cambios habrá hecho *git merge* con la rama master y finalmente *git push* al repositorio GitLab.

De esta manera, nosotros hacemos *git pull* de la rama principal y comprobamos que efectivamente nuestros cambios aparecen incluidos en la rama *main* (o master).

```
(base) brosaaf-l3209-pc29:~/practica-isi$ git pull
Actualizando a3ed40d..87c02f7
Fast-forward
 members.txt | 1 +
 1 file changed, 1 insertion(+)
(base) brosaaf-l3209-pc29:~/practica-isi$ cat members.txt
Juan Antonio Ortega Aparicio
Belén Rosa Alonso
```

Ilustración 6: Git pull de rama principal y comprobar cambios incluidos desde terminal

## 1.2. Acepta cambios de un compañero de tu equipo

A continuación, un compañero de nuestro equipo realizará cambios en el repositorio del equipo igual que nosotros hicimos en el apartado 1.1. A pesar de tener permiso de modificar directamente la rama principal mediante *git push*, el compañero subirá una rama creada desde su entorno local para que nosotros integremos los cambios en la rama *main*.

1. Al igual que nosotros en el apartado 1.1, nuestro compañero hizo clone del proyecto, creó una rama local en su repositorio local llamada *pablobv/nuevos.requisitos*, modificó el archivo *members.txt* añadiendo su nombre, hizo commit y finalmente, hizo *git push* de la rama en repositorio grupal GitLab. El commit indicando la modificación en los archivos está en la url <https://gitlab.etsit.urjc.es/grupo-isi/practica-isi/-/commit/069a535c4c4ab3f5e71c5f6ada01e4f3c3e80cbc>.
2. A continuación, nosotros creamos un remoto con *git remote add username\_compañero origin*, siendo origin el repositorio origen de GitLab y hacemos *git fetch* de la rama de nuestro compañero.

```
(base) brosaaf-l3202-pc05:~/practica-isi$ git remote add pablobv origin
fatal: remoto pablobv ya existe.
(base) brosaaf-l3202-pc05:~/practica-isi$ git fetch origin pablobv/nuevos.requisitos
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (3/3), 361 bytes | 45.00 KiB/s, listo.
Desde https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi
* branch                pablobv/nuevos.requisitos -> FETCH_HEAD
87c02f7..069a535 pablobv/nuevos.requisitos -> origin/pablobv/nuevos.requisitos
```

Ilustración 7: Creación de remoto y fetch de la rama de nuestro compañero desde el terminal

Darse cuenta que, si hubiésemos hecho *git pull*, se habría hecho un merge automático con los cambios introducidos por nuestro compañero y la rama principal del repositorio.

3. Creamos una rama local vinculada a la rama remota para revisar las modificaciones introducidas en la rama de nuestro compañero.

```
(base) brosaa@f-l3202-pc05:~/practica-isi$ git branch mirarcambios origin/pablov
v/nuevos.requisitos
Rama 'mirarcambios' configurada para hacer seguimiento a la rama remota 'pablov
v/nuevos.requisitos' de 'origin'.
(base) brosaa@f-l3202-pc05:~/practica-isi$ git checkout mirarcambios
Cambiado a rama 'mirarcambios'
Tu rama está actualizada con 'origin/pablov/nuevos.requisitos'.
(base) brosaa@f-l3202-pc05:~/practica-isi$ git branch
  brosaa/nuevos-requisitos
  master
* mirarcambios
```

Ilustración 8: Creación de rama local para hacer seguimiento de rama remota desde terminal

4. Finalmente hacemos *git merge* de la rama que con los cambios y la rama *main* del repositorio local y hacemos *git push* hacia el repositorio grupal para actualizar la rama principal en GitLab.

Aparece un commit indicando el merge con la rama *mirarcambios* en el repositorio GitLab, accesible desde la url <https://gitlab.etsit.urjc.es/grupo-isi/practica-isi/-/commit/657d0a639b2124ae21152d7c7600b194e7dcfa39>.



Ilustración 9: Commit del merge con la rama local desde GitLab

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

### 2.1. Sube cambios que deben ser aceptados por otro compañero de otro equipo

En este caso, vamos a trabajar en el repositorio GitLab de un compañero de otro equipo. El objetivo es realizar cambios en los archivos del repositorio de un compañero externo y que me acepten dichos cambios, en el que no tengo permiso de escritura, es decir, no puedo incluir cambios directamente haciendo un *git push*.

1. Hacemos fork del repositorio grupal del equipo de mi compañero, Samuel Martín Ortega. Para ello, vamos a la página GitLab de su repositorio <https://gitlab.etsit.urjc.es/grupodeisi/practica-isi> y realizamos un fork para crear un repositorio copia del repositorio del otro equipo.

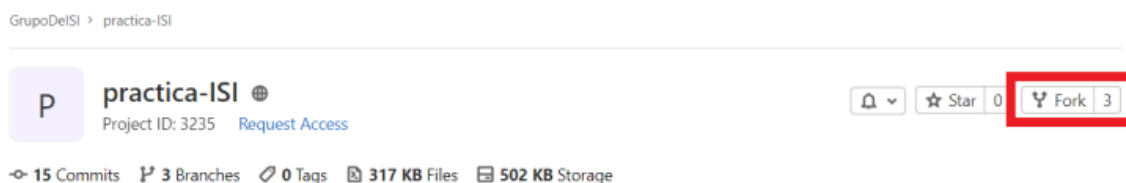


Ilustración 10: Repositorio GitLab del grupo externo

Dicho repositorio copia estará ahora en nuestra cuenta personal de GitLab, por lo que podremos modificarla sin restricciones. Asegurarse de crear el repositorio público para los pasos posteriores o tendremos que modificarlo en *Settings > General > Visibility, project features, permissions > Public visibility*.

Nuestro repositorio ahora está en la URL <https://gitlab.etsit.urjc.es/brosaa/practica-isi-samuel>.

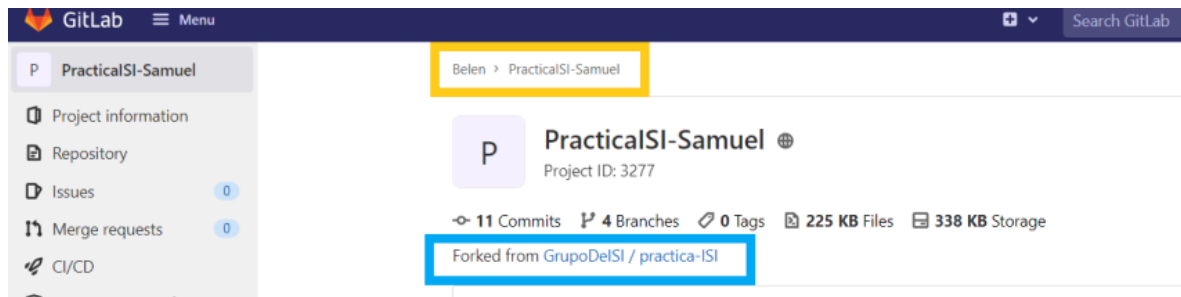


Ilustración 11: Repositorio personal forkeado del repositorio externo

2. Como el repositorio es nuestro, lo clonamos en nuestro equipo local usando el comando `git clone` <https://gitlab.etsit.urjc.es/brosaa/practica-isi-samuel> en una terminal.

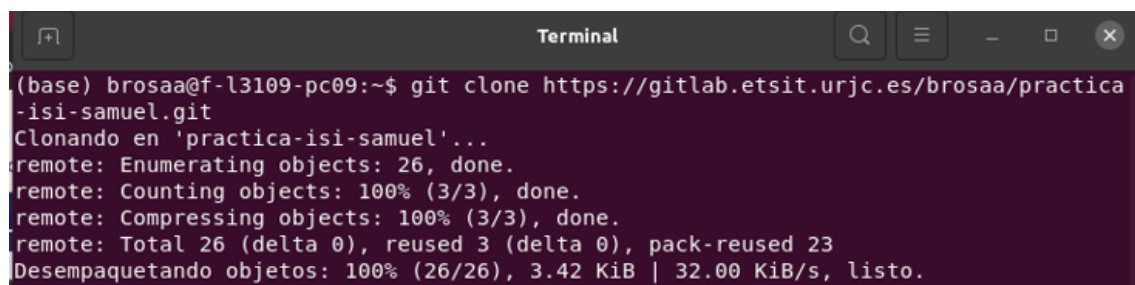


Ilustración 12: Clonado del repositorio forkeado

3. Entramos en el repositorio clonado y creamos una nueva rama local mediante el comando `git branch` ***brosaa/nuevos-requisitos*** y entramos en ella mediante el comando `git checkout` ***brosaa/nuevos-requisitos***.



Ilustración 13: Creación de nueva rama local

Podemos comprobar las ramas mediante el comando `git branch`, donde veremos las ramas existentes y en color verde y un asterisco (\*) la rama donde estamos situados.



4. Modificamos el archivo `members.txt` añadiendo una línea con “Belén Rosa Alonso – Grupo externo” para incluirme como miembro, pero dejando claro que soy de otro equipo. A continuación, hago `git add` y añado un commit con `git commit -m`.

```
(base) brosaaf-l3109-pc09:~/practica-isi-samuel$ cat memebbers.txt
Samuel Martín Ortega
Paula Pulido Praena
Amelia Sánchez Aparicio
Belén Rosa Alonso - Grupo externo
(base) brosaaf-l3109-pc09:~/practica-isi-samuel$ git add --all
(base) brosaaf-l3109-pc09:~/practica-isi-samuel$ git commit -m "Belén añadida (externa)"
```

Ilustración 14: Archivo modificado y creación de commit de la modificación

5. Hacemos push desde la rama `brosaa/nuevos-requisitos` donde hemos hecho las modificaciones hacia nuestro origen, es decir, nuestro proyecto forkeado mediante el comando `git push origin brosaaf-nuevos-requisitos` para que los cambios y los commit creados en local se añadan en el repositorio de GitLab.

```
(base) brosaaf-l3109-pc09:~/practica-isi-samuel$ git push origin brosaaf-nuevos-requisitos
Username for 'https://gitlab.etsit.urjc.es': brosaaf
Password for 'https://brosaaf@gitlab.etsit.urjc.es':
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 397 bytes | 397.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
remote:
remote: To create a merge request for brosaaf/nuevos-requisitos, visit:
remote:   https://gitlab.etsit.urjc.es/brosaaf/practica-isi-samuel/-/merge_requests/new?merge_request%5Bsource_branch%5D=brosaaf%2Fnuevos-requisitos
remote:
To https://gitlab.etsit.urjc.es/brosaaf/practica-isi-samuel.git
```

Ilustración 15: Push de la rama local desde terminal

Desde la propia terminal, ya nos sugiere crear un **merge request** para que un miembro del equipo externo revise y haga merge de mis cambios en el repositorio del equipo, donde recordemos que yo no puedo modificar directamente.

El commit en mi proyecto forkeado se encuentra en <https://gitlab.etsit.urjc.es/brosaaf/practica-isi-samuel/-/commit/ab71bf91bc6c80710eb0b6477d19e28d81ff7752>.

6. Realizamos un merge request a través de la página del repositorio forkeado de GitLab para que se ingresen las modificaciones. Para ello, me dirijo al repositorio del equipo externo del cual hice el fork y creo un merge request en la página merge requests situada en el menú de la izquierda de GitLab (cuadro naranja, ilustración 16). Nos aseguramos de pedir un merge desde la rama `brosaa/nuevos-requisitos` hacia la rama `main` (o master) del repositorio grupal (cuadro verde, ilustración 16).

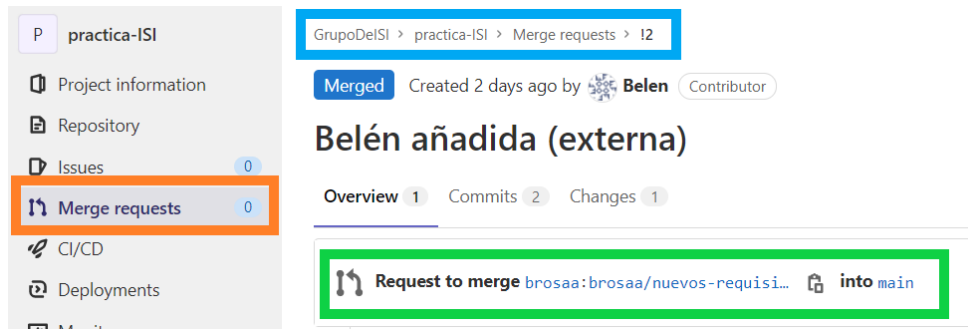


Ilustración 16: Merge request al repositorio grupal de GitLab

Nos fijamos en que los merge requests se identifican con un número incremental, en nuestro caso es el número 2 (cuadro azul, ilustración 16).

7. Vemos en la página de merge request que un miembro del otro equipo ha revisado mis cambios, pero no está de acuerdo con ellos, por lo que incluye un comentario en mi merge request para que haga una nueva modificación.

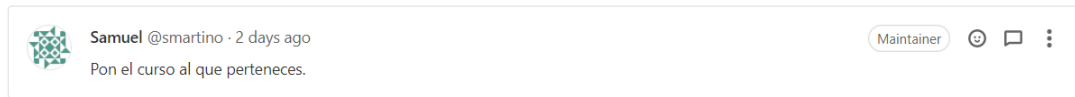


Ilustración 17: Comentario a mi merge request de un miembro del otro equipo

8. Realizamos las modificaciones en proyecto local que nos pide el compañero, desde la rama *brosaa/nuevos-requisitos*, los añadimos, creamos un commit y hacemos push de todo ello.

```
(base) brosaa@f-l3109-pc09:~/practica-isi-samuel$ git add --all
(base) brosaa@f-l3109-pc09:~/practica-isi-samuel$ git branch
* brosaa/nuevos-requisitos
main
(base) brosaa@f-l3109-pc09:~/practica-isi-samuel$ git commit -m "Añadido el curso"
[brosaa/nuevos-requisitos ec9b307] Añadido el curso
1 file changed, 1 insertion(+), 1 deletion(-)
(base) brosaa@f-l3109-pc09:~/practica-isi-samuel$ git push origin brosaa/nuevos-requisitos
Username for 'https://gitlab.etsit.urjc.es': brosaa
Password for 'https://brosaa@gitlab.etsit.urjc.es':
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (3/3), listo.
```

Ilustración 18: Commit y push de las modificaciones desde terminal

Este commit, situado en la URL [https://gitlab.etsit.urjc.es/grupodeisi/practica-isi/-/merge\\_requests/2/diffs?commit\\_id=ec9b3071a78824306dcc50e23fff364321230f9d](https://gitlab.etsit.urjc.es/grupodeisi/practica-isi/-/merge_requests/2/diffs?commit_id=ec9b3071a78824306dcc50e23fff364321230f9d), aparecerá de forma automática en la página del merge request como respuesta al comentario de mi compañero.

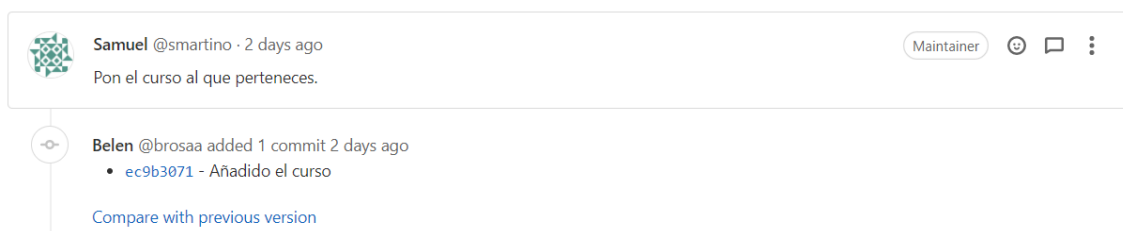


Ilustración 19: Nuevo commit como respuesta al comentario en el merge request

9. Mi compañero se habrá descargado en su repositorio local mi rama con las modificaciones añadidas, si está de acuerdo hará merge en el *main* del repositorio y finalmente *git push* hacia GitLab.

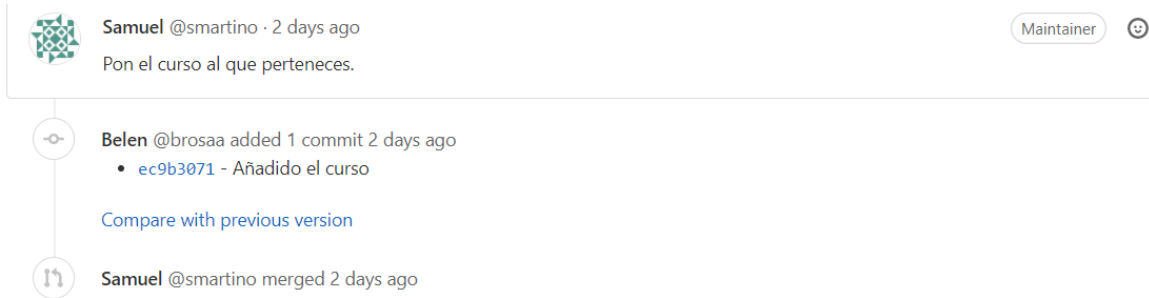


Ilustración 20: Hilo final de comentarios en el merge request tras el merge

A continuación, yo hago *git pull* de la rama principal del repositorio del otro equipo y los cambios aparecerán incluidos en la rama *main*.

```
(base) brosaa@f-l3109-pc09:~/practica-isi-samuel$ git branch
brosaa/nuevos-requisitos
* main
(base) brosaa@f-l3109-pc09:~/practica-isi-samuel$ git pull https://gitlab.etsit.urjc.es/grupodeisi/practica-isi
warning: redirigiendo a https://gitlab.etsit.urjc.es/grupodeisi/practica-isi.git/
Desde https://gitlab.etsit.urjc.es/grupodeisi/practica-isi
* branch      HEAD      -> FETCH_HEAD
Actualizando 06cc172..ec9b307
Fast-forward
 memebbers.txt | 2 ++
 1 file changed, 2 insertions(+)
(base) brosaa@f-l3109-pc09:~/practica-isi-samuel$ cat memebbers.txt
Samuel Martín Ortega
Paula Pulido Praena
Amelia Sánchez Aparicio
Belén Rosa Alonso - Grupo externo, curso 2017
```

Ilustración 21: Git pull de la rama main del repositorio grupal en terminal

En la ilustración 21 comprobamos que efectivamente aparecen las modificaciones con mi nombre y mi curso en el archivo de members desde mi rama principal local.

10. Hacemos *git push* de la rama *main* local hacia la rama *main* del repositorio en GitLab forkeado del repositorio grupal y finalmente tendremos la misma versión de rama *main* con mis modificaciones tanto en local, como en el repositorio GitLab forkeado como en el repositorio GitLab grupal externo.

```
(base) brosaa@f-l3109-pc09:~/practica-isi-samuel$ git push https://gitlab.etsit.urjc.es/brosaa/practica-isi-samuel
Username for 'https://gitlab.etsit.urjc.es': brosaa
Password for 'https://brosaa@gitlab.etsit.urjc.es':
warning: redirigiendo a https://gitlab.etsit.urjc.es/brosaa/practica-isi-samuel.git/
Total 0 (delta 0), reusado 0 (delta 0)
To https://gitlab.etsit.urjc.es/brosaa/practica-isi-samuel
 06cc172..ec9b307 main -> main
```

Ilustración 22: Git push de la rama main local al repositorio GitLab forkeado desde terminal

## 2.2. Acepta cambios de un compañero de otro equipo

A continuación, un compañero externo a nuestro equipo querrá realizar cambios en el repositorio de nuestro equipo igual que nosotros hicimos en el apartado 2.1. Como el compañero no tiene permiso de escritura en nuestro repositorio, nos hará un **merge request** para que nosotros integremos los cambios en la rama *main*.



1. Nuestro compañero hizo fork de nuestro proyecto, creó una rama local en su repositorio local llamada *smartino/Nuevos\_Requisitos*, hizo *git push* en su repositorio GitLab forkeado y finalmente nos hizo el merge request pidiéndonos aceptar sus cambios en la rama *master* (o *main*) de nuestro repositorio grupal en la url [https://gitlab.etsit.urjc.es/grupo-isi/practica-isi/-/merge\\_requests/3](https://gitlab.etsit.urjc.es/grupo-isi/practica-isi/-/merge_requests/3).

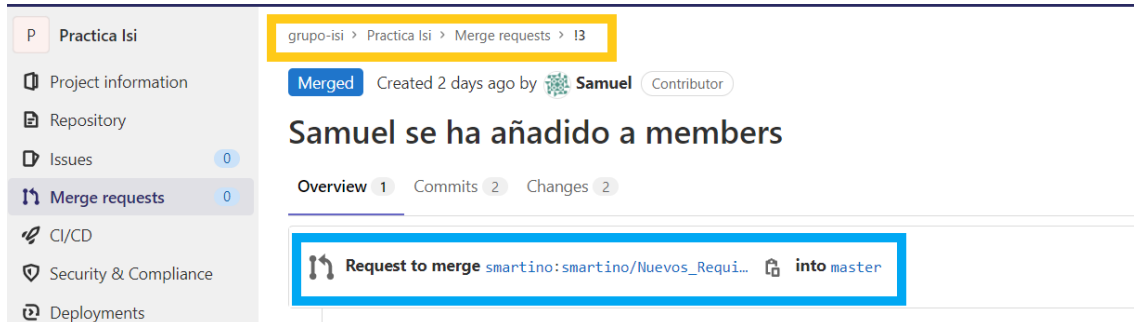


Ilustración 23: Merge request creado por compañero externo a nuestro repositorio grupal

2. Miramos los cambios introducidos por nuestro compañero y hacemos una petición para modificar los archivos mediante un comentario en el propio merge request. A continuación, nuestro compañero realiza en el proyecto local los cambios que le pedí en el comentario, hace commit y *git push*, se puede ver en la url [https://gitlab.etsit.urjc.es/grupo-isi/practica-isi/-/merge\\_requests/3/diffs?commit\\_id=c7f49583e7901e8a73f9561545d96e34f1f72d77](https://gitlab.etsit.urjc.es/grupo-isi/practica-isi/-/merge_requests/3/diffs?commit_id=c7f49583e7901e8a73f9561545d96e34f1f72d77). Dicho commit aparece de forma automática como respuesta a mi comentario.

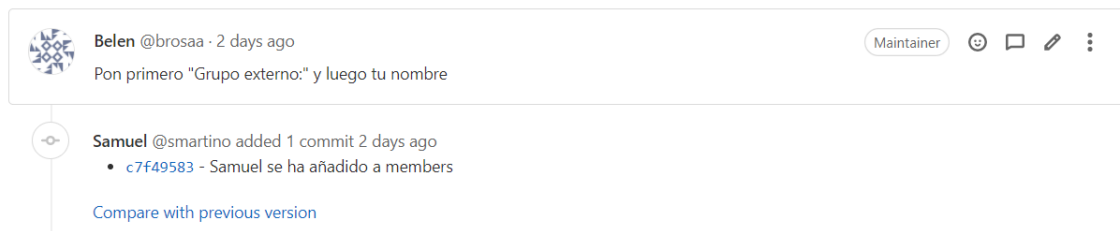


Ilustración 24: Commit de mi compañero externo con los cambios que le pedí en el comentario del merge request

3. A continuación, me bajo en mi repositorio clonado local la rama *smartino/Nuevos\_Requisitos* de mi compañero mediante un *git fetch* (si hiciera *git pull* del repositorio, se haría *git fetch* + *git merge* automáticamente).

Para hacer *git fetch* de un merge request hay que tener en cuenta el identificador del merge request mencionado en el punto 2.1.6 y usar el comando *git fetch origin merge-requests/mergerequestID/head:nombre-de-rama*, quedando en nuestro caso como en la ilustración 25.

```
(base) brosaa@f-l3109-pc09:~/practica-isi$ git fetch origin merge-requests/3/head:smartino/Nuevos_Requisitos
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (7/7), 1.18 KiB | 19.00 KiB/s, listo.
Desde https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi
* [nueva referencia] refs/merge-requests/3/head -> smartino/Nuevos_Requisitos
```

Ilustración 25: Git fetch de la rama de un merge request en terminal

Si estamos de acuerdo con las modificaciones, hacemos merge para incluir los cambios en la rama *main* de mi repositorio clonado en local y miramos el flujo de trabajo para comprobar que se incluyen las modificaciones con el comando `git log --graph` en el terminal.

```
(base) brosaaf-l3109-pc09:~/practica-isi$ git merge smartino/Nuevos_Requisitos
Actualizando b606e33..c7f4958
Fast-forward
 .members.txt.swp | Bin 0 -> 12288 bytes
 members.txt      | 2 +-
 2 files changed, 1 insertion(+), 1 deletion(-)
 create mode 100644 .members.txt.swp
(base) brosaaf-l3109-pc09:~/practica-isi$ git log --graph
* commit c7f49583e7901e8a73f9561545d96e34f1f72d77 (HEAD -> master, smartino/Nuevos_Requisitos)
   Author: Samuel Martin Ortega <s.martino.2016@alumnos.urjc.es>
   Date:   Thu Nov 11 15:38:19 2021 +0100

       Samuel se ha añadido a members

* commit 9c33905272ffda323ccf233bb426a113378677a0
```

Ilustración 26: Merge del merge request en el terminal

Finalmente, hacemos `git push` de la rama *main* del proyecto local al repositorio grupal de GitLab, de esta manera habremos incluido los cambios de un editor externo.

```
(base) brosaaf-l3109-pc09:~/practica-isi$ git push
Username for 'https://gitlab.etsit.urjc.es': brosaaf
Password for 'https://brosaaf@gitlab.etsit.urjc.es':
Enumerando objetos: 9, listo.
Contando objetos: 100% (9/9), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (7/7), listo.
Escribiendo objetos: 100% (7/7), 1.20 KiB | 306.00 KiB/s, listo.
Total 7 (delta 0), reusado 0 (delta 0)
remote:
remote: Project 'c.borao.2017/practica-isi' was moved to 'grupo-isi/practica-isi'.
remote:
remote: Please update your Git remote:
remote:
remote:   git remote set-url origin https://gitlab.etsit.urjc.es/grupo-isi/practica-isi.git
remote:
remote:
To https://gitlab.etsit.urjc.es/c.borao.2017/practica-isi.git
b606e33..c7f4958 master -> master
```

Ilustración 27: Git push de la rama main local al repositorio GitLab en terminal

Comprobamos que el merge aparece de forma automática en el hilo del merge request del repositorio GitLab grupal.

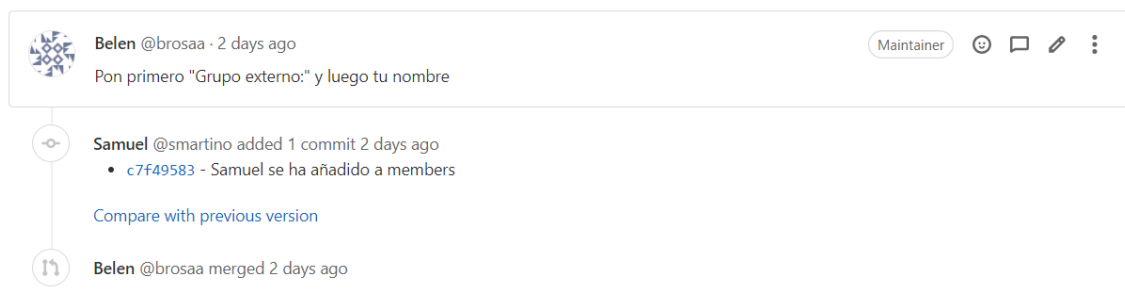


Ilustración 28: Hilo del merge request del repositorio grupal en GitLab