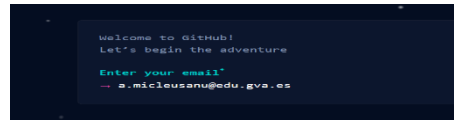


Actividad 8. Navegar y descubrir las posibilidades que permite GitHub para proyectos colaborativos

Creación de una cuenta en GitHub

Paso 1. Ir a la página <https://github.com/>

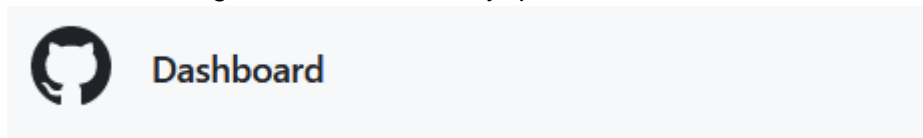
Paso 2. Sign Up



Paso 3. Introduce un nombre, email y se verifica el correo para validar.

Paso *** 3.1. Si tenéis ya una cuenta creada se puede seguir utilizando sin ningún problema.

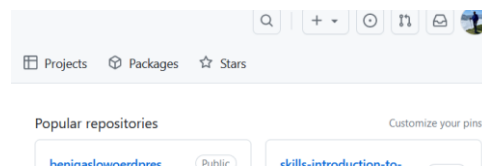
Paso 4. Visión general de la cuenta y que ocurre cuando nos creamos la cuenta.



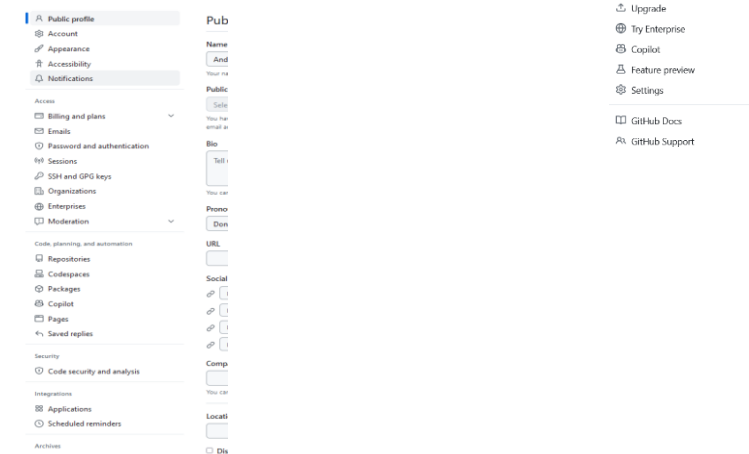
Una visión general de vuestra actividad en la plataforma.

Paso 5.

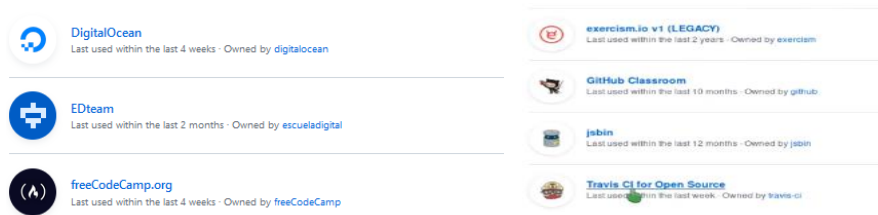
Perfil: el botón - + , notificaciones, etc.



Paso 6. Configuración de la cuenta:
apretando sobre el avatar o el logo del perfil,
se desplegará:
se puede ver si tenemos seguidores estrella
y en los setings
tenemos el acceso a la configuración completa:

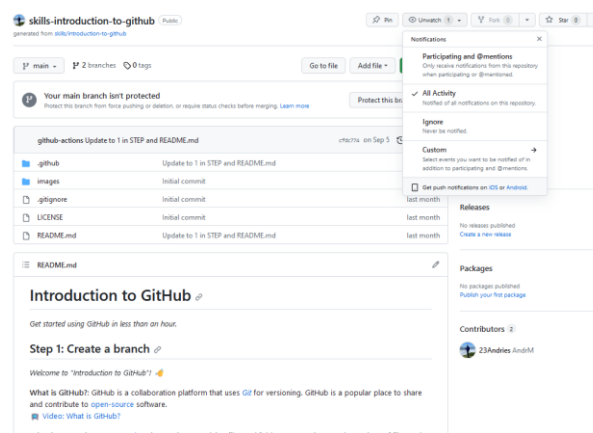


Paso 7. Comprobamos las filas de: emails, repositorios, aplicaciones, que están autorizadas, etc.



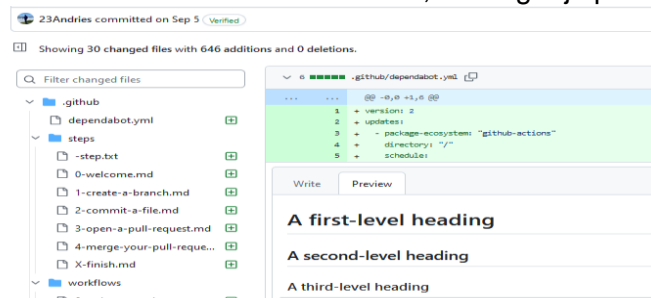
Paso 8. Seguidores, estrellas y comentarios de los repositorios

El seguimiento de un repositorio se puede observar accediendo y apretando sobre un repositorio, se puede ver en watch, se puede contactar con el desarrollador pinchando sobre ... y de esta manera nos hacemos seguidor de.

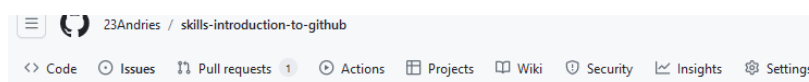


El tema de los comentarios, podemos hacerlo haciendo clic sobre un commit en concreto o sobre determinadas líneas de código se crea un comentario y dicho comentario crea un hilo y determinadas personas interesadas en proyectos podrían participar.

El formato de comentarios es el formato Markdown, un lenguaje plano, de formato ligero,

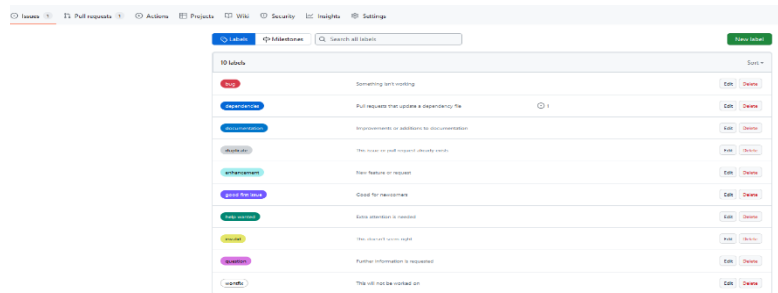


Paso 9. Temas como Issues, ... nos permiten publicar tareas, llevar bugs, o cualquier otra cosa relacionada con el repositorio, etc.

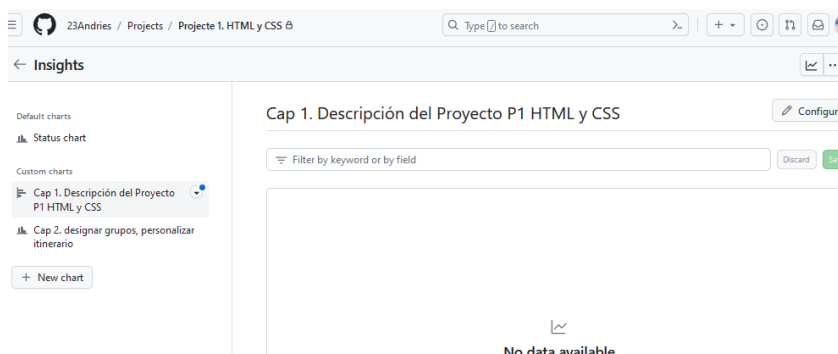


Gestión de tareas y proyectos.

Paso 10. Cerrar los Issues a través de los cambios de código., tenemos etiquetas (labels) creadas y que podemos asociarlas a las Issues.



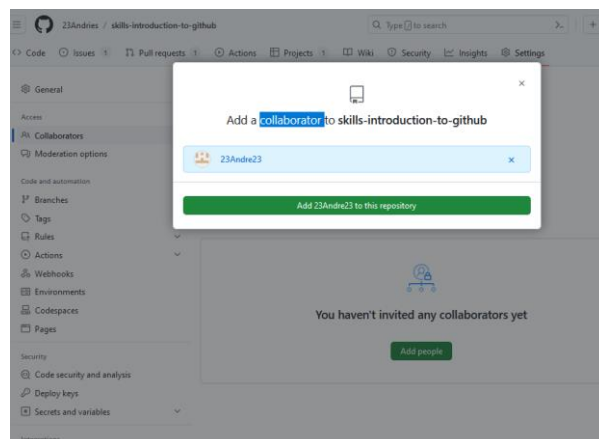
Paso 11. Proyectos. Gestionar tareas grandes de grupo o similares.



Paso 12. Añadir colaboradores a un proyecto.

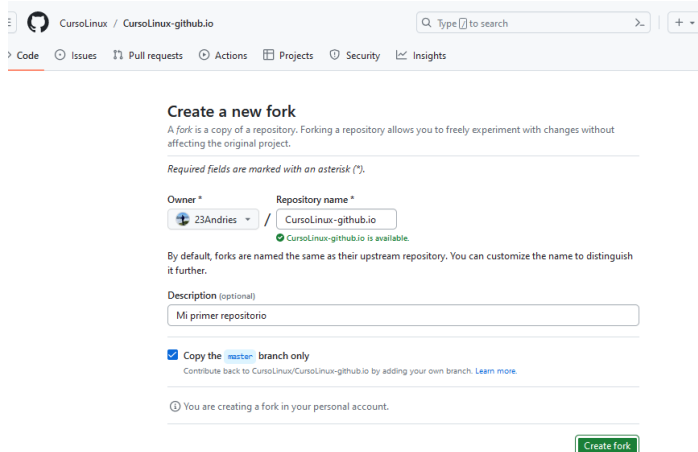
Primero localizamos un repositorio dentro de nuestra lista y lo seleccionamos. Apretamos en Settings y le damos los permisos de colaborador, a través del usuario o del correo electrónico.

Paso 13. Los permisos del colaborador son idénticos al propietario salvo unos cuantos, es decir tienen acceso a comunicar y a los miembros del equipo de desarrollo.



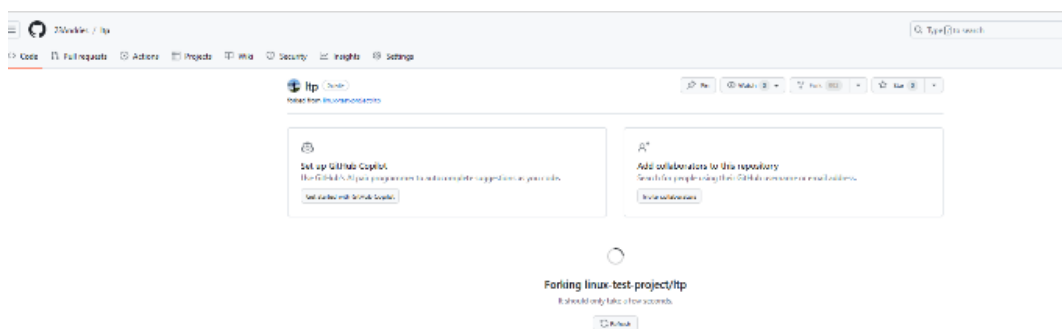
Paso 14. El sistema de fork y/o pull request. Este sistema consiste

Por ejemplo, se selecciona un repositorio y se quiere clonar a través de la opción fork. Una copia del dicho repositorio en mi cuenta.



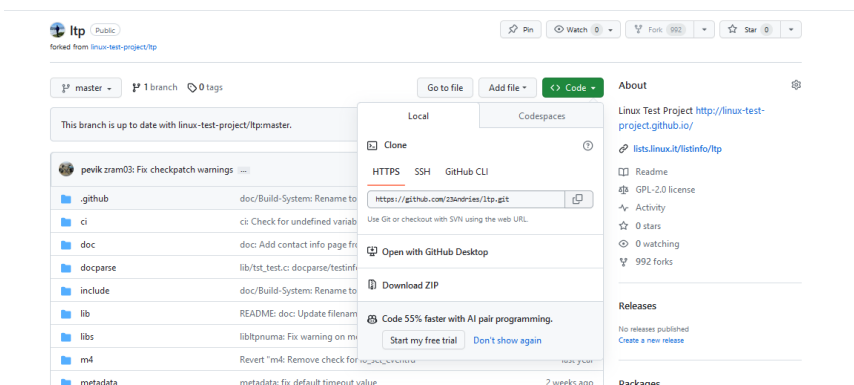
The screenshot shows the 'Create a new fork' page on GitHub. At the top, it says 'Create a new fork' and explains that a fork is a copy of a repository. Below this, it states 'Required fields are marked with an asterisk (*)'. The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' is set to '23Andries' and the 'Repository name' is 'CursoLinux-github.io'. A message below says 'CursoLinux-github.io is available'. There is a 'Description (optional)' field with the text 'Mi primer repositorio'. A checkbox 'Copy the master branch only' is checked. At the bottom, there is a 'Create fork' button.

otro ejemplo para realizar el fork:



si queremos contribuir a través de pull request. Cambiamos en el repositorio que tenemos en nuestro repositorio local, en principio se debe pedir permiso al dueño para que acepte nuestros cambios y los adjunte a su versión.

Paso 15. Por ejemplo, una vez que tenemos el repositorio clonado en nuestra cuenta. Iniciamos el git clone

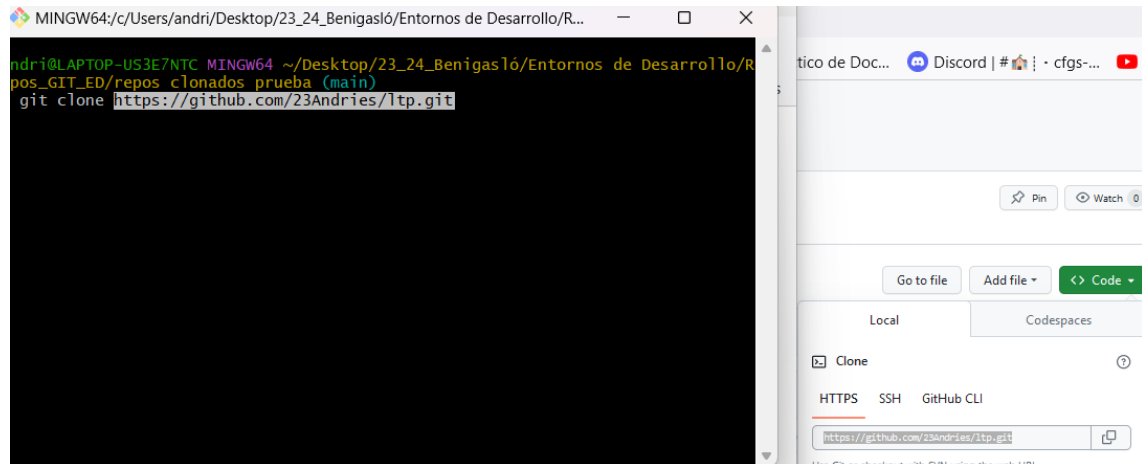


Paso 16. En nuestra terminal

¡Importante! Primero creamos un repositorio (carpeta), después abrimos la terminal desde esta carpeta (por ejemplo, con el botón derecho del ratón).

Teniendo la ruta, comprobamos que es la nuestra ruta de nuestro equipo.

Escribimos **git clone** y seguido la ruta (por ejemplo, esta ruta la obtenemos después de realizar el fork en mi equipo. <https://github.com/23Andries/ltp.git>)



Esta puede ser una variante de respuesta en la terminal después de clonar.

```
andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasló/Entornos de Desarrollo/R...
epos_GIT_ED/repos clonados prueba (main)
$ git clone https://github.com/23Andries/ltp.git
Cloning into 'ltp'...
remote: Enumerating objects: 214351, done.
remote: Counting objects: 100% (6348/6348), done.
remote: Compressing objects: 100% (680/680), done.
remote: Total 214351 (delta 5714), reused 5811 (delta 5665), pack-reused 208003
Receiving objects: 100% (214351/214351), 62.74 MiB | 10.91 MiB/s, done.
Resolving deltas: 100% (161136/161136), done.
Updating files: 100% (8794/8794), done.
```

Captura de la terminal

```
andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba (main)
$ git clone https://github.com/23Andries/ltp.git
Cloning into 'ltp'...
remote: Enumerating objects: 214351, done.
remote: Counting objects: 100% (6348/6348), done.
remote: Compressing objects: 100% (680/680), done.
remote: Total 214351 (delta 5714), reused 5811 (delta 5665), pack-reused
208003
Receiving objects: 100% (214351/214351), 62.74 MiB | 10.91 MiB/s, done.
Resolving deltas: 100% (161136/161136), done.
updating files: 100% (8794/8794), done.

andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba (main)
$ ls
ltp/

andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba/ltp (master)
$ ls
COPYING          TODO          docparse/      metadata/      testcases/
Containerfile    VERSION      include/       pan/           testscripts/
IDcheck.sh*     build.sh*    lib/           runltp*        tools/
INSTALL         ci/          libs/          runtest/       utils/
Makefile        configure.ac ltpmenu*       scenario_groups/ ver_linux*
README.md       doc/         m4/            scripts/

andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba/ltp (master)
$ cd doc/

andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba/ltp/doc (master)
$ touch lista1.txt

andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba/ltp/doc (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  lista1.txt

nothing added to commit but untracked files present (use "git add" to track)

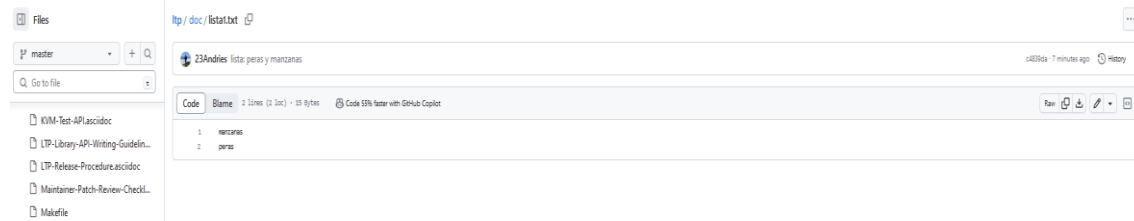
andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba/ltp/doc (master)
$ git add .

andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba/ltp/doc (master)
$ git commit -m " lista: peras y manzanas"
[master c4839da1f] lista: peras y manzanas
1 file changed, 2 insertions(+)
create mode 100644 doc/lista1.txt

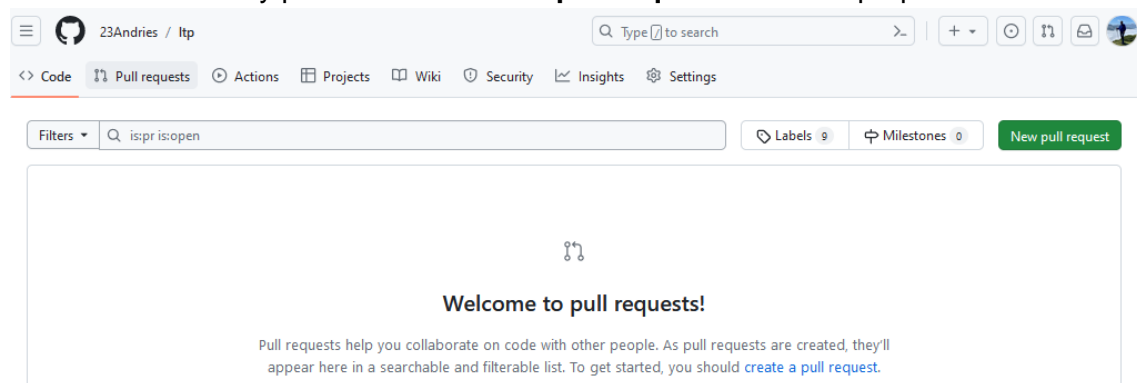
andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigasl6/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba/ltp/doc (master)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 353 bytes | 353.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/23Andries/ltp.git
50a587989..c4839da1f master -> master
```

```
andri@LAPTOP-US3E7NTC MINGW64 ~/Desktop/23_24_Benigas16/Entornos de
Desarrollo/Repos_GIT_ED/repos clonados prueba/ltp/doc (master)
$
```

Si venimos en el repositorio GitHub y vemos dentro de la carpeta .doc localizamos el archivo lista1.txt

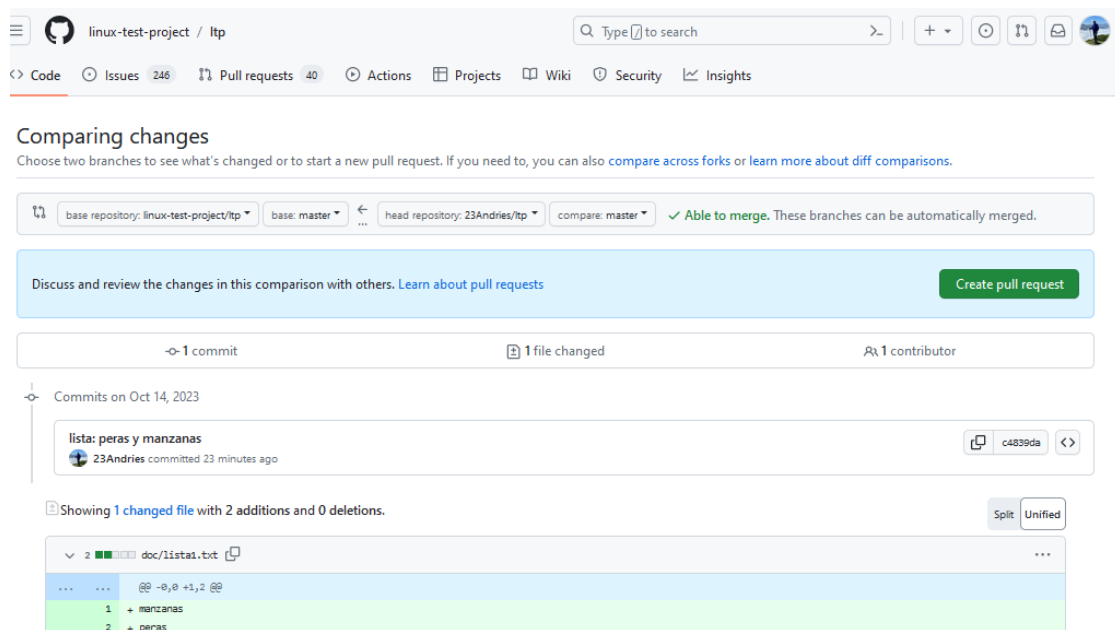


Tenemos cambios y podemos realizar un **pull request** desde su propio menú:



Apretamos el botón para **Create pull request**

Se hace



En el caso que no se llega a un acuerdo para realizar la fusión se cierra con **Close pull request**.

The screenshot shows a GitHub pull request interface. At the top, there is a green box with a status message: "Continuous integration has not been set up" and "This branch has no conflicts with the base branch". Below this, there is a green button labeled "Merge pull request" and a link "or view command line instructions.". Below the merge button, there is a text area for leaving a comment, a "Close pull request" button, and a "Comment" button. At the bottom, there is a "ProTip!" message: "Add .patch or .diff to the end of URLs for Git's plaintext views."

Al contrario, después de hacer el pull request, y si tenemos el acuerdo de la cuenta tendremos. Que significa se ha aceptado nuestra petición con un resultado positivo, en términos de git es: un pase en el origen **linux-text-project/ltip** se han traído los cambios desde **23Andries/ltip**.

This is a smaller version of the screenshot above, showing the same GitHub pull request interface with the "Merge pull request" and "Close pull request" buttons.

¡Importante! si no somos el dueño del repositorio del destino no podemos controlar si nos acepta o no nos rechaza la fusión (**merge**), además se han de tener en cuenta posibles conflictos que se deben solucionar antes, y esto puede llevar tiempo para resolver. Es decir se han de poner de acuerdo las partes implicadas.

