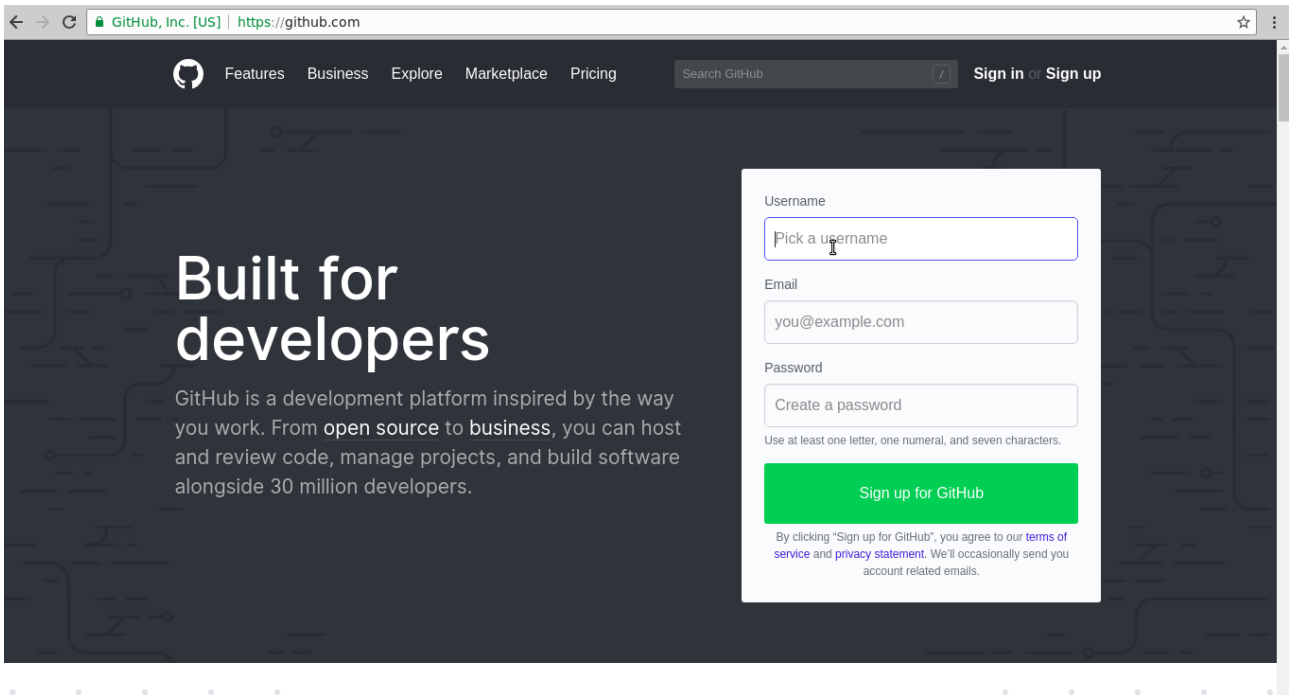
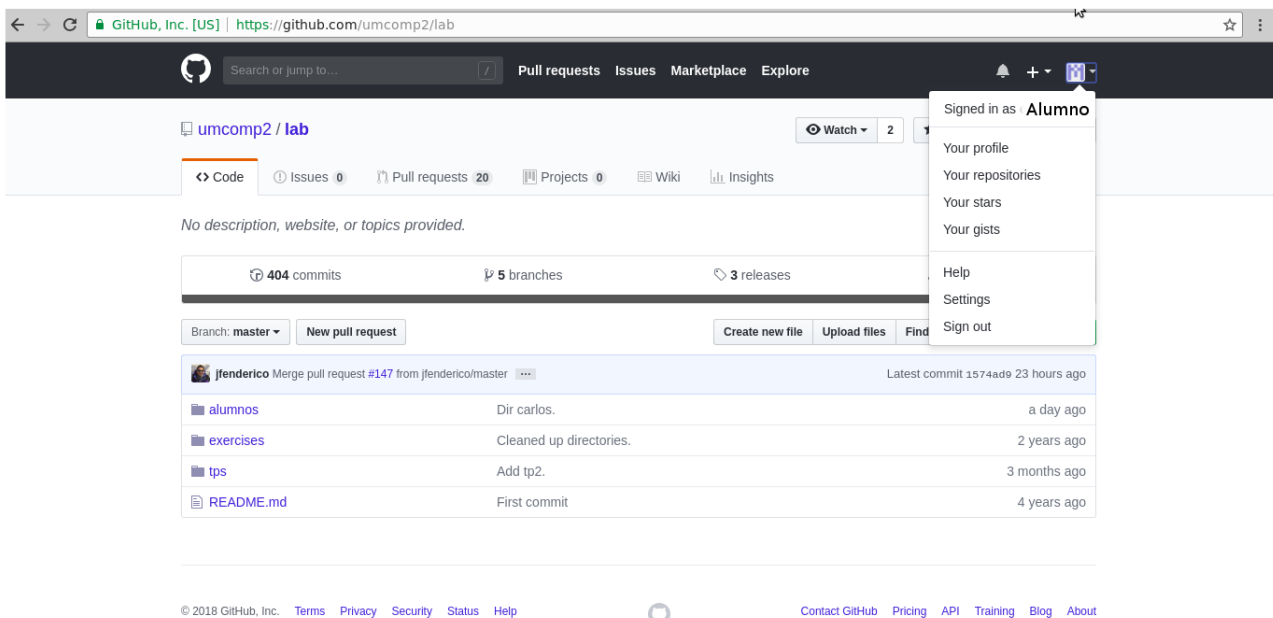


Tutorial para entrega de Prácticos usando Github

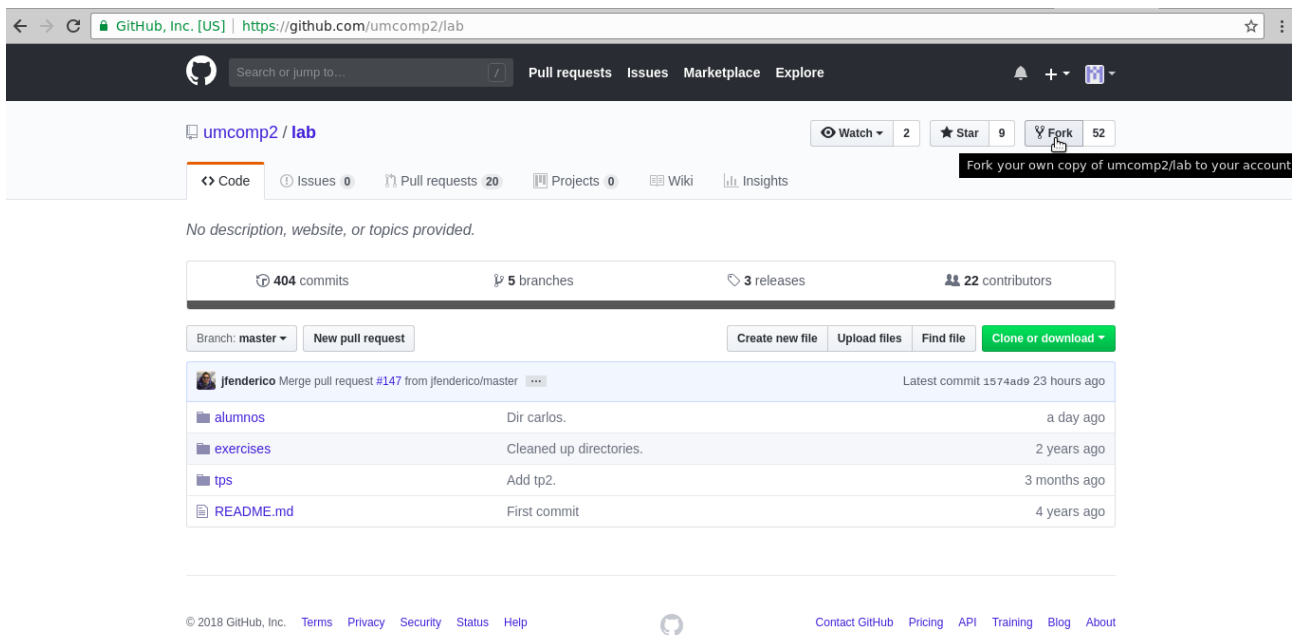
1 – Crear una cuenta en github, en caso de no tener, accediendo con un browser a <https://github.com/>



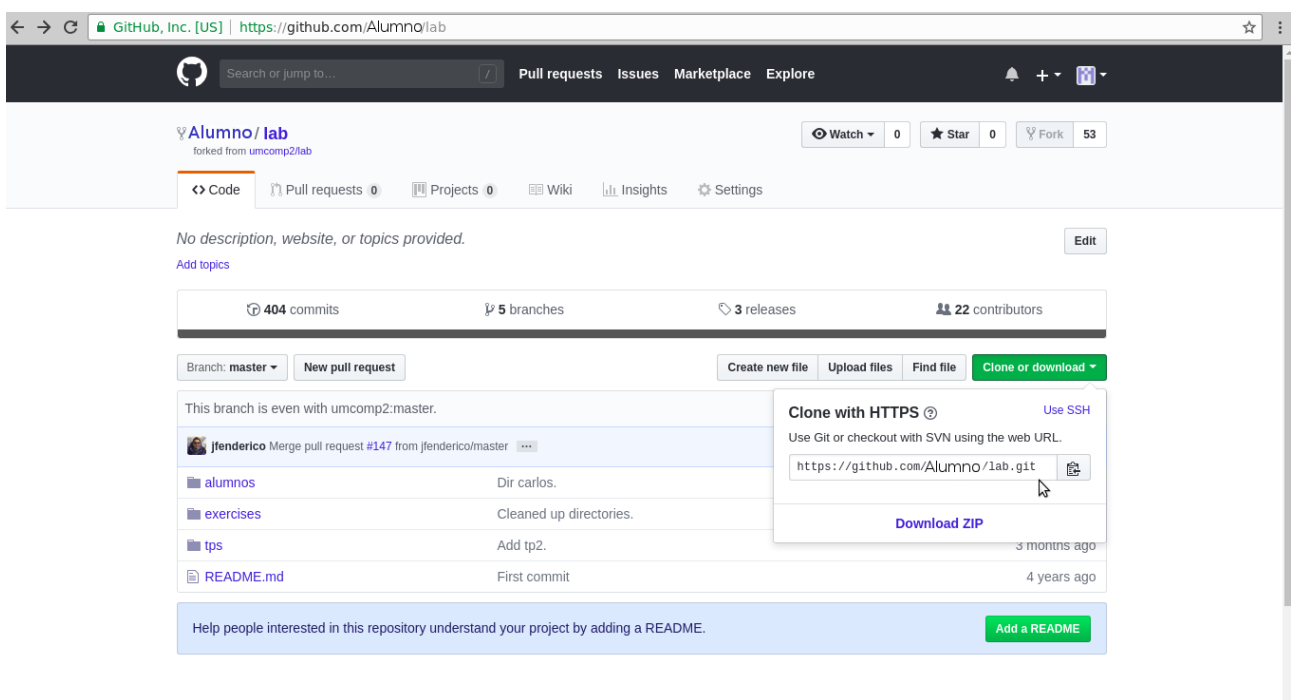
Una vez que ingresas con TU usuario (Alumno) a github, acceder al link <https://github.com/umcomp2/lab/>



2- Hacer un fork del proyecto “lab” de umcomp2 en tu usuario, dándole al botón como en el gráfico



Luego si vuelves al home de tu usuario (Alumno), vas a ver el nuevo proyecto, como en la siguiente figura:



3- Ahora estas en condiciones de clonarlo en tu computadora, para poder trabajar.

Creá un directorio (compu2, por ejemplo) con el siguiente comando:

```
alumno@note:~/ $ mkdir compu2
alumno@note:~/compu2 $ cd compu2
```

para luego clonar el proyecto. En caso de no tener instalado github en tu computadora, esto es lo que tenes que ejecutar para instalarlo (bajo gnu/ubuntu)

```
alumno@note:~/compu2 $ sudo apt-get install git
```

luego ejecutar

```
alumno@note:~/compu2$ git clone https://github.com/Alumno/lab.git
Username for 'https://github.com': alumno@gmail.com
Password for 'https://alumno@gmail.com@github.com':
```

Este comando va a pedirte tu nombre de usuario y clave antes de descargar el proyecto localmente

Configurá tu nombre y correo en git, para que cada vez que enviás modificaciones figure tu nombre

```
git config --user.email "alumno@mail.com"
git config --user.name "Alumno"
```

Luego con el siguiente comando, podes ver el estado, que debería ser el mismo que figura acá abajo

```
alumno@note:~/compu2$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

Antes de hacer los prácticos, ingresá al directorio alumnos y creá un directorio con tu legajo + “-“ + nombre + “-” + Apellido . Dentro de ese nuevo directorio, creá un archivo llamado contacto, con tu nombre, apellido + “, “ + dirección de correo electrónico

```
alumno@note:~/compu2$ cd alumnos
alumno@note:~/compu2/alumnos$ mkdir 0000-alumno-prueba
alumno@note:~/compu2/alumnos$ cd 0000-alumno-prueba
alumno@note:~/compu2/alumnos/0000-alumno-prueba$ echo "alumno prueba, alumno_prueba@mail.com" > contacto
alumno@note:~/compu2/alumnos/0000-alumno-prueba$ cd ../../
```

nuevamente en el directorio compu2, vemos el status. (Ahora dirá que localmente tenemos un archivo sin versionar)

```
alumno@note:~/compu2$ 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)

  alumnos/0000-alumno-prueba/

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

4- Agregamos el archivo para versionar

```
alumno@note:~/compu2$ git add alumnos/0000-alumno-prueba/
```

Revisamos nuevamente el estado:

```
alumno@note:~/compu2$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   alumnos/0000-alumno-prueba/contacto
```

5- Actualizar la versión (comitear).Hacerlo SIEMPRE desde el directorio base compu2:

```
alumno@note:~/compu2$ git commit -m "comit local"
```

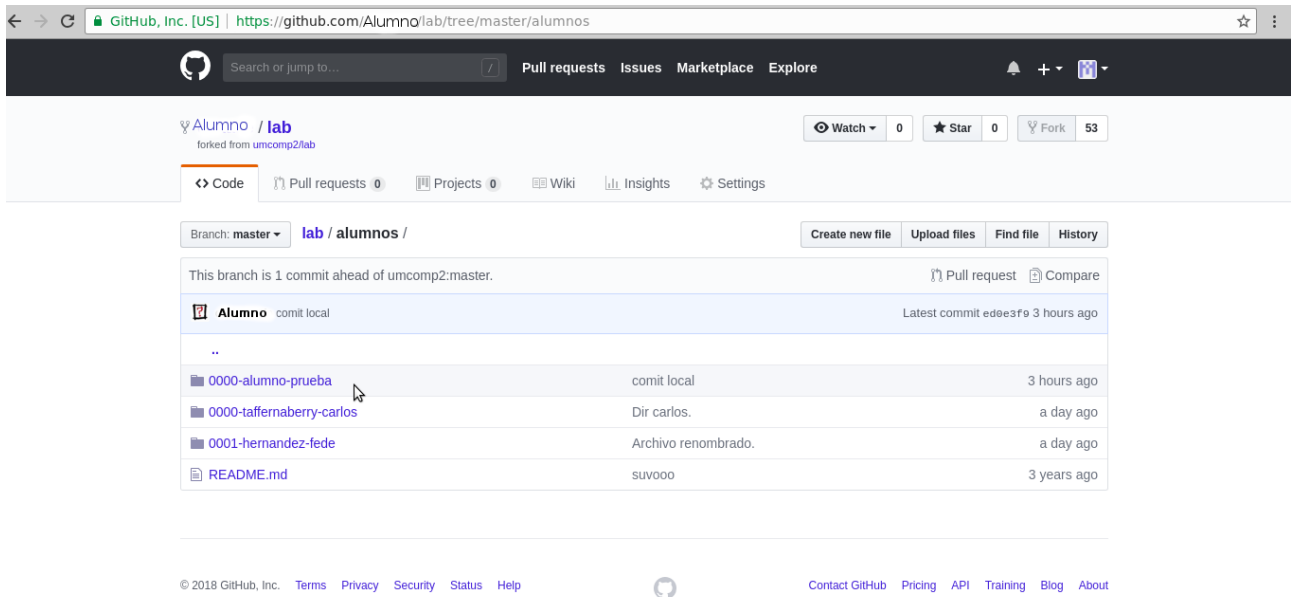
6- Actualizar el repositorio remoto (en github) , Hacerlo SIEMPRE desde el directorio base compu2:

```
alumno@note:~/compu2$ git push origin master
Username for 'https://github.com': alumno@gmail.com
```

Password for 'https://alumno@gmail.com@github.com':

```
Counting objects: 5, done.  
Delta compression using up to 2 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (5/5), 489 bytes | 0 bytes/s, done.  
Total 5 (delta 1), reused 0 (delta 0)  
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To https://github.com/alumno/lab.git  
1574ad9..ed0e3f9 master -> master
```

si volvemos a acceder con el browser a <https://github.com/alumno/lab>, verificaremos que ya está visible el nuevo directorio creado, como se ve en la siguiente figura.



7- Finalmente, para solicitar que se apliquen los cambios al repositorio oficial de la materia (umcomp2), acceder con el browser a la url <https://github.com/umcomp2/lab/> y solicitar un “pull request”, como se ve en la siguiente imagen.

GitHub, Inc. [US] | <https://github.com/umcomp2/lab>

Search or jump to... Pull requests Issues Marketplace Explore

umcomp2 / lab Watch 2 Star 9 Fork 53

Code Issues 0 Pull requests 20 Projects 0 Wiki Insights

No description, website, or topics provided.

404 commits 5 branches 3 releases 22 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

jfenderico Merge pull request #147 from jfenderico/master Latest commit 1574ad9 a day ago

alumnos	Dir carlos.	a day ago
exercises	Cleaned up directories.	2 years ago
tps	Add tp2.	3 months ago
README.md	First commit	4 years ago

© 2018 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

<https://github.com/umcomp2/lab/pull/new/master>

Asegurarse de seleccionar “Compare across forks”, como se puede ver

GitHub, Inc. [US] | <https://github.com/umcomp2/lab/compare?expand=1>

Search or jump to... Pull requests Issues Marketplace Explore

umcomp2 / lab Watch 2 Star 9 Fork 53

Code Issues 0 Pull requests 20 Projects 0 Wiki Insights

Compare changes

Compare changes across branches, commits, tags, and more below. If you need to, you can also [compare across forks](#).

base: master compare: master

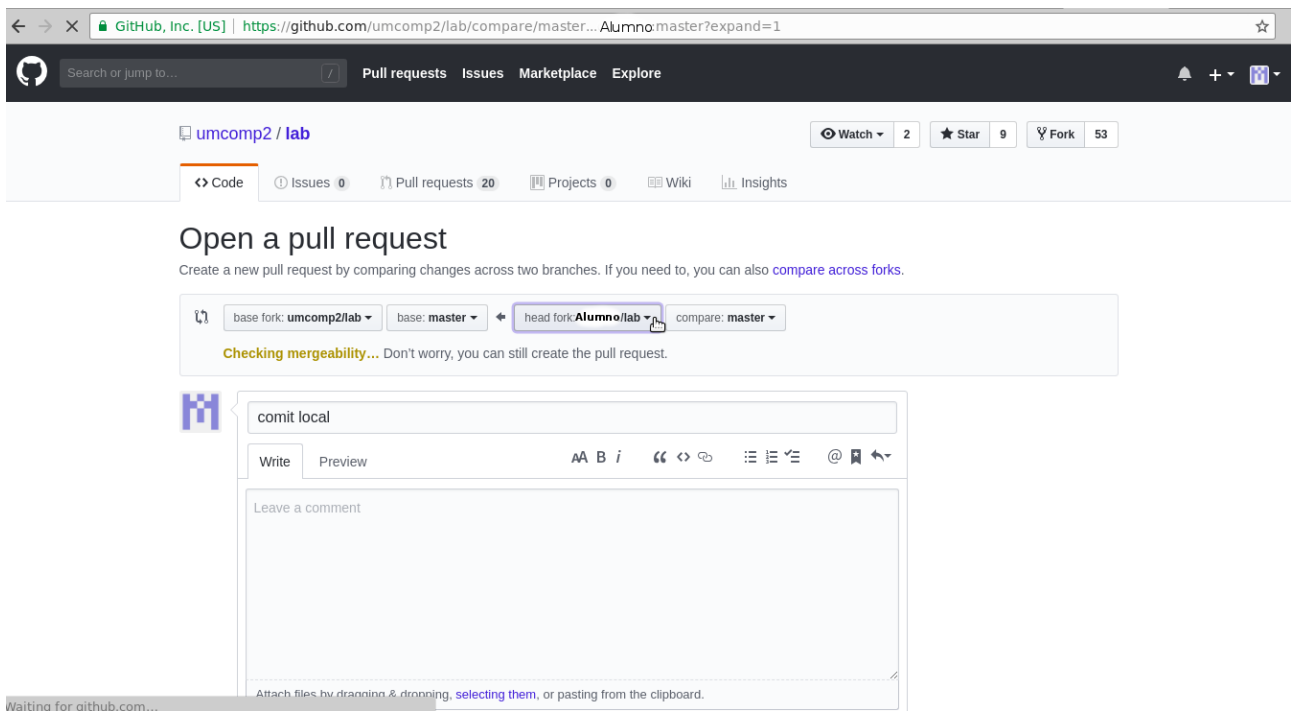
Create pull request Choose different branches or forks above to discuss and review changes.

Compare and review just about anything

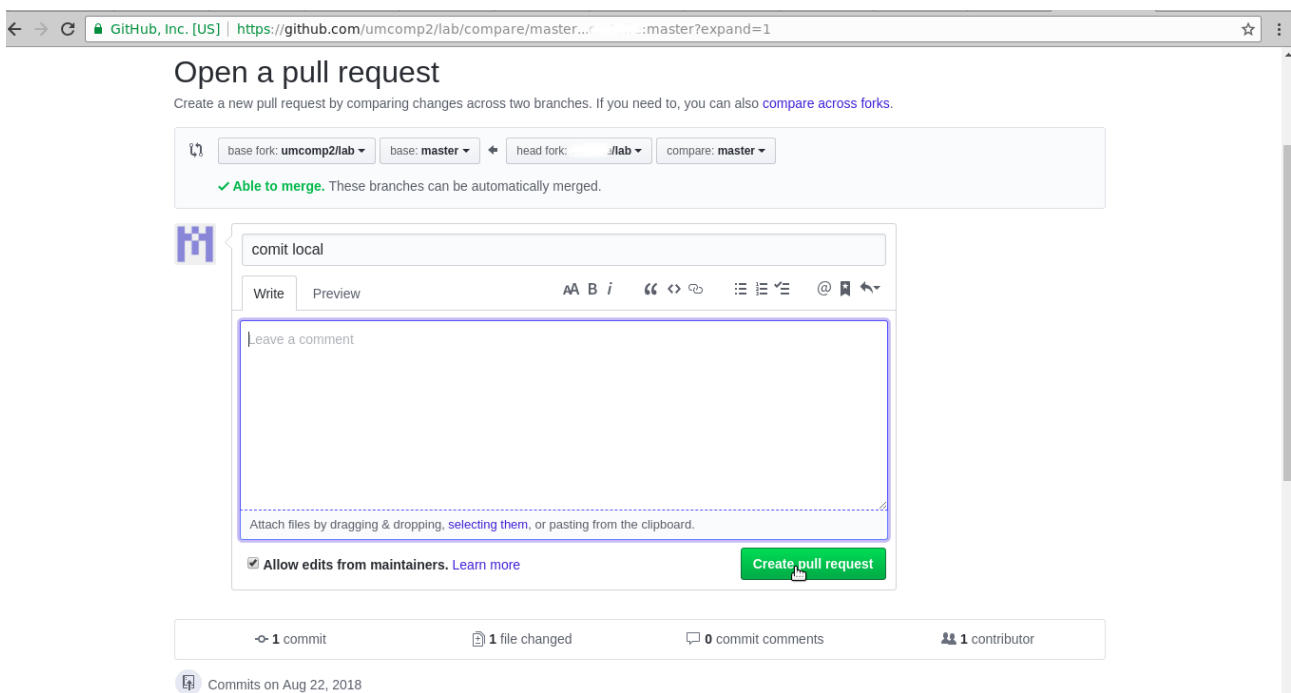
Branches, tags, commit ranges, and time ranges. In the same repository and across forks.

EXAMPLE COMPARISONS	
2017	on Apr 20
2017-tp2	on May 23, 2017

Asegurarse de seleccionar el “Head Fork” el de ustedes, en este caso, Alumno, como se ve a continuación



por último darle “Create pull request”, lo que avisará a los profes, que hay cambios que tenemos que revisar.



8- Cada vez que hagan un práctico, creen un directorio en lab/tps de su repositorio local (computadora), escriban el código, versionenlo con git add, git commit y git push . Cuando esté terminado, deben repetir el paso 7 – Pedido de pull request

9- Para el caso que quieran actualizar su repositorio local (Computadora) con los nuevos enunciados de los prácticos desde el repositorio oficial (umcomp2), deben asegurarse de tener configurados los repositorios remotos como figuran a continuación, usando el siguiente comando

```
alumno@note:~/compu2$ git remote -v
oficial    https://github.com/umcomp2/lab.git (fetch)
oficial    https://github.com/umcomp2/lab.git (push)
```

```
origin    https://github.com/alumno/lab.git (fetch)
origin    https://github.com/alumno/lab.git (push)
```

En caso de no existir los remotos “oficial”, agregarlos usando el siguiente comando:

```
alumno@note:~/compu2$ git remote add comp2 https://github.com/umcomp2/lab.git
```

10 – Para actualizar el repo local, ejecutar :

```
alumno@note:~/compu2$ git pull oficial master
```