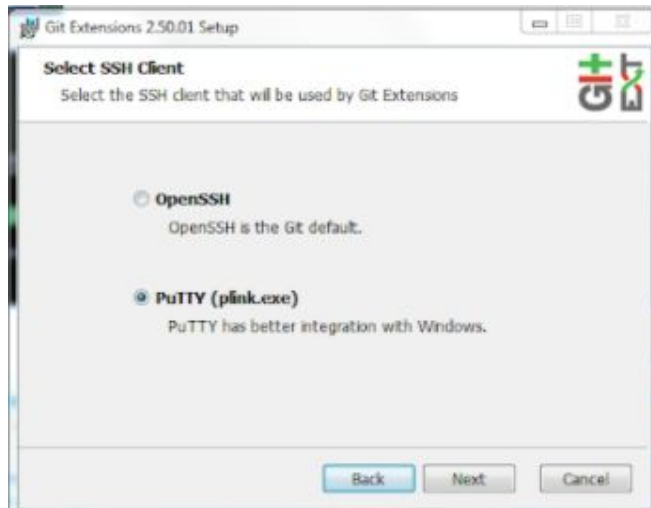


Chicos, yo tuve problemas con “**Putty**” cuando configure la clave SSH, les recomiendo desinstalen Git Extensions, y cuando lo vuelvan a instalar elijan **OpenSSH**



### Configurar clave SSH

- 1) Abrimos Git Extension, click en el icono de Git bash
- 2) `cd ~`
- 3) `mkdir .ssh`

*(para crear el key con el cuál nos va a identificar GitHub, lo creamos en la maquina, y luego ya no tenemos que poner las credenciales)*

- 4) `cd ~/.ssh`

*(entramos a esa carpeta)*

- 5) `ssh-keygen -t rsa -C "emma.pusso@gmail.com"`

*(generamos un key ssh con la dirección que pusimos en Giy y Git Extensions)*

- 6)

//ENTER (default name) //

//ENTER (no passphrase) //

### Aparece:

```
Your identification has been saved in /c/Users/Manu/.ssh/id_rsa.
Your public key has been saved in /c/Users/Manu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:aSjJ+2G0ht47ZnRTtAEps20WzX3k9T3vGTE1FySCses emma.pusso@gmail.com
The key's randomart image is:
+---[RSA 2048]---+
|  .o .o. .o. |
| o . o.o o .o |
| B o * . . .+ |
| ,*.o.+ .+ .+ |
| o+...So      = |
| .oo. . .    . |
| o..o. E     .o |
| ..== .      . |
| ..++oo      . |
+---[SHA256]-----+
```

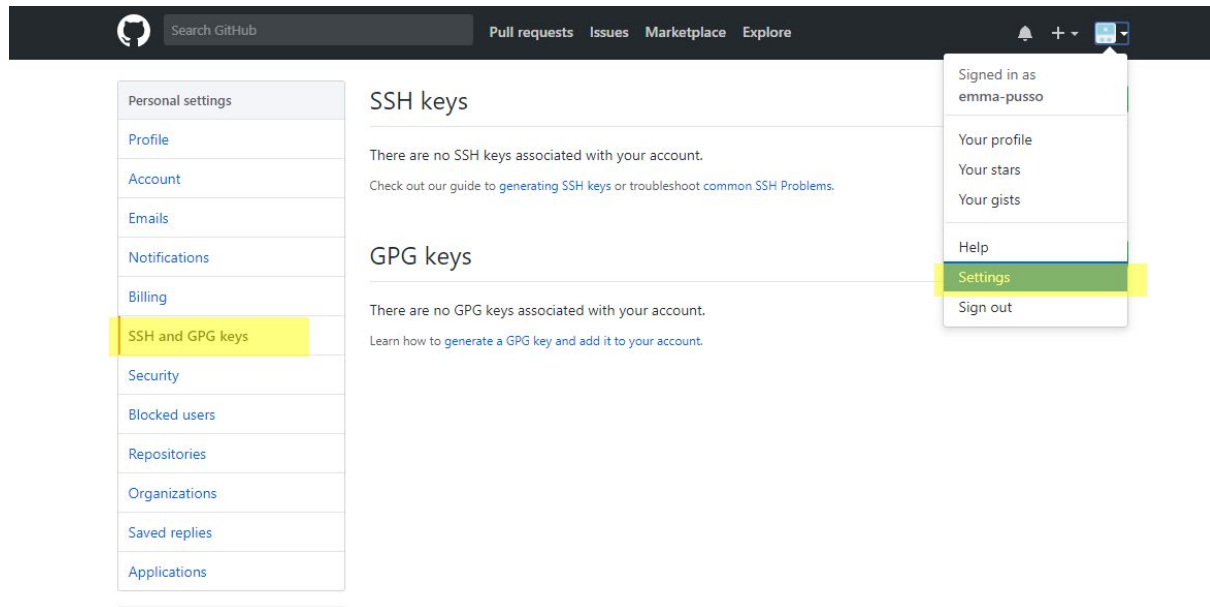
7) eval \$(ssh-agent -s)

8) ssh-add ~/.ssh/id\_rsa

9) clip < ~/.ssh/id\_rsa.pub

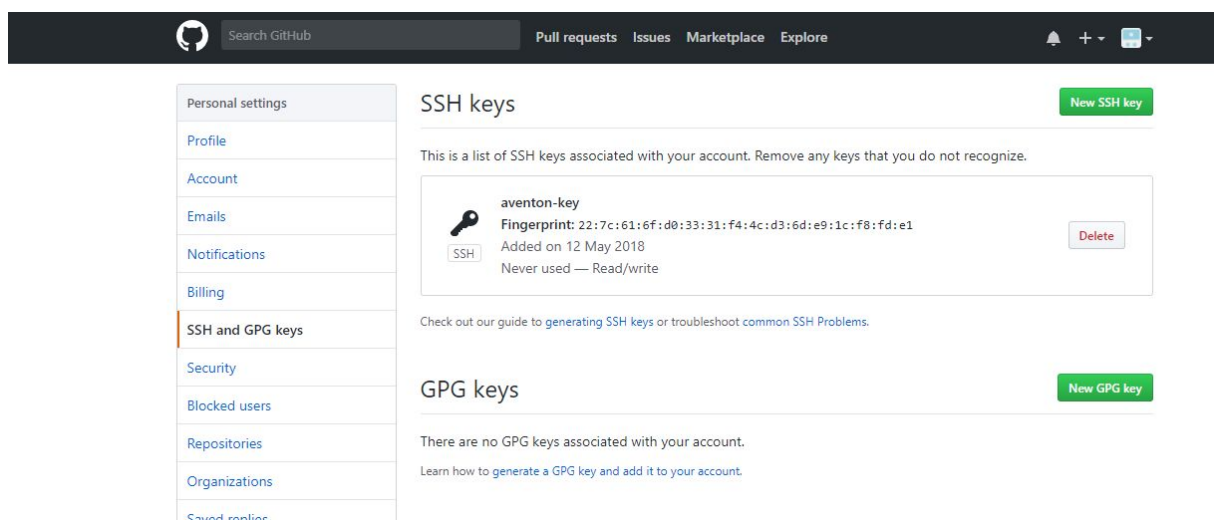
(Copia el contenido de la clave **id\_rsa.pub** para luego pegar en github)

10) Agregar la key en GitHub:



Click en [New SSH key]

Pegar la clave (copiada en el paso 9), y un nombre y guardar. Les va a quedar como:



## **Configurar Plugin de GitHub de GitExtensions**

- 1) En Git Extensions: Plugins -> Git Hub
- 2) Ingresar credenciales de github
- 3) Click en [Authotize application]
- 4) Aparece un pop up de confirmación con el mensaje “Successfully retrieved O Auth token”

## **Configurar repos remotos**

### 1) Configurar **Origin**

- a) En Git Extensios, ir a **Repository > Remote Repository**

Name: origin

Url: [git@github.com:emma-pusso/IngSoft-grupo34.git](https://git@github.com:emma-pusso/IngSoft-grupo34.git)  
(reemplazar emma-pusso por su usuario de github)

Click en [Save changes]

### 2) Configurar **Upstream**

- a) En Git Extensios, ir a **Repository > Remote Repository**

- b) Click en el icono **[+]** para añadir un nuevo repo remoto

Name: upstream

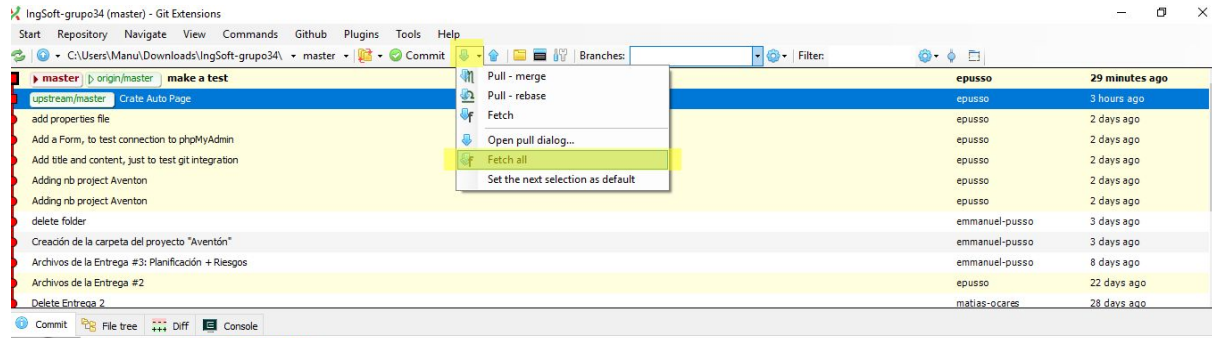
Url: [git@github.com:emmanuel-pusso/IngSoft-grupo34.git](https://git@github.com:emmanuel-pusso/IngSoft-grupo34.git)  
(este es igual para todos, no lo cambien)

- c) Aparece un mensaje de si quieren configurar este repo como el default para el Push and Pull, denle click en [Si]

Click en [Save changes]

## Flujo de trabajo

### 1) Fetch all:

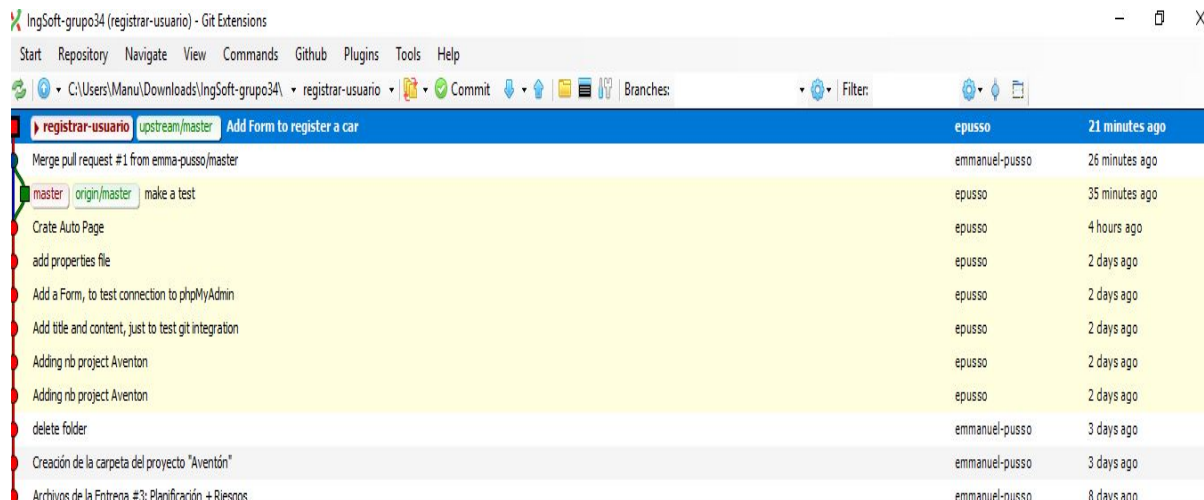


Me va a traer los cambios que haya en el repo origin y ustream, pero no me integra sino que los guarda en el índice de cambios.

2) Crear una rama por cada nueva funcionalidad (historia de usuario) que vaya a desarrollar, a partir de **Upstream/master**.

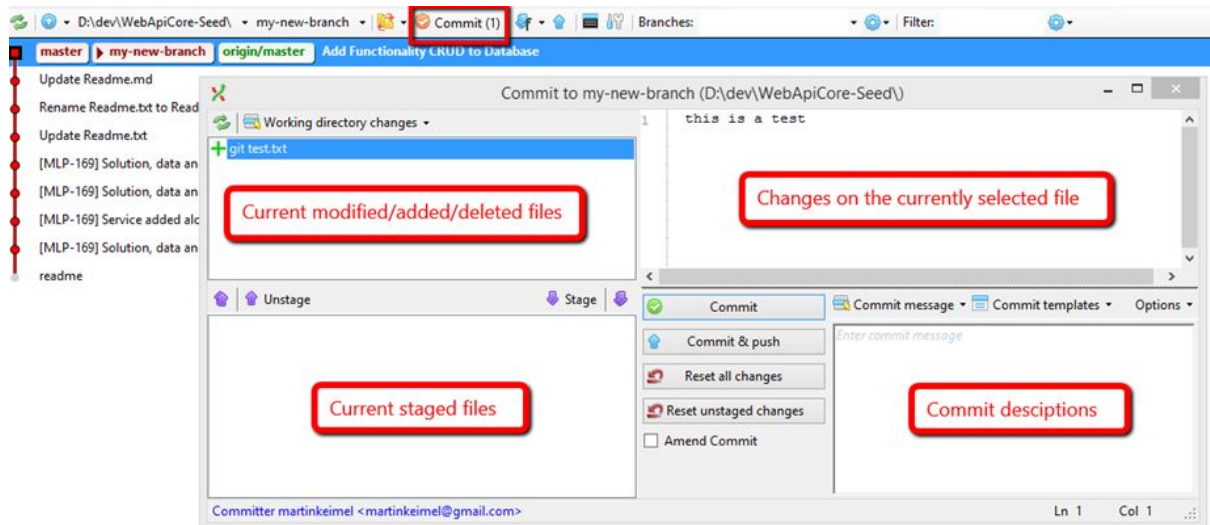
Click derecho sobre la rama, y seleccionar “**New local branch**”

Ingresar un nombre descriptivo, el nombre de la HU que estemos desarrollando, por ejemplo: registrar-usuario



3) Desarrollo la funcionalidad, puedo ir haciendo commits parciales.

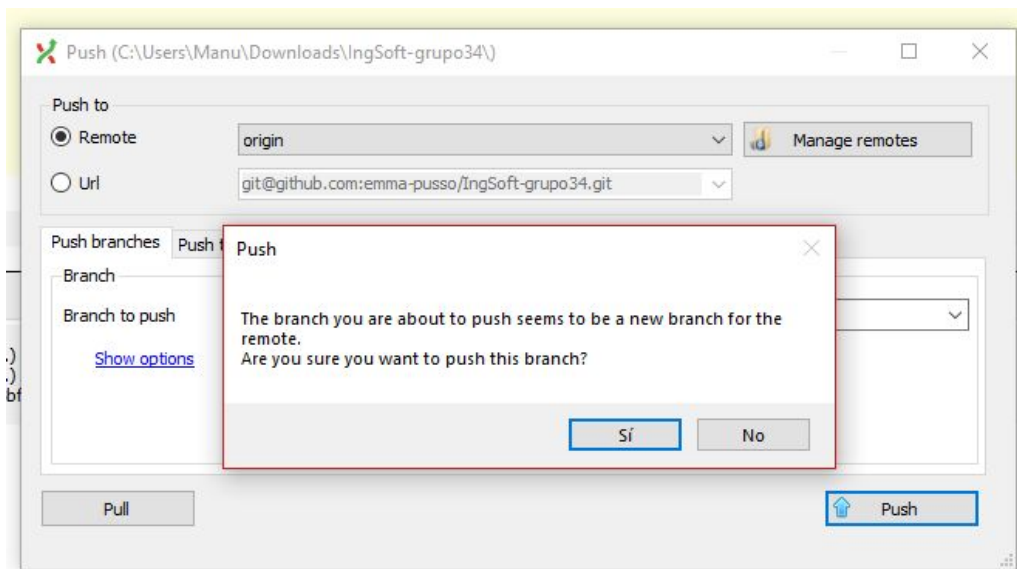
Hasta este momento todos los cambios están en la rama local **registrar-usuario**. Todavía no existe la rama remota **origin/registrar-usuario** (se va a crear cuando hago un push)



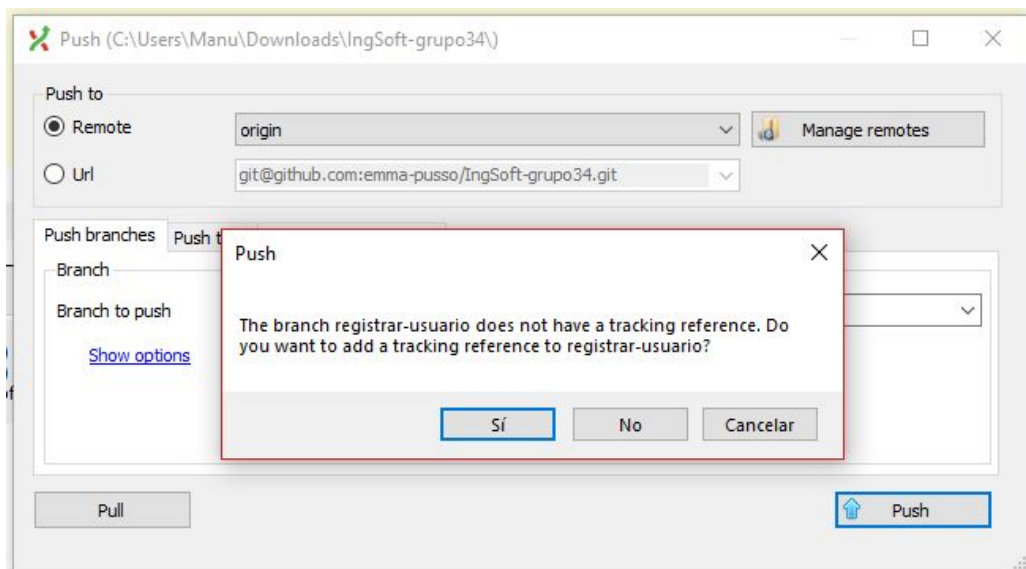
4) Desde el momento que cree la rama hasta que realice los commits, pudo haber nuevos cambios en el repo upstream/master, por lo cuál **antes de hacer el push** tengo que traerme esos nuevos cambios, integrarlos, si hay conflictos los resuelvo en local, y finalmente hacer el push:

- Fetch All
- Si hay cambios **upstream/master**
  - Click derecho sobre upstream/master, y seleccionar **Rebase current branch on upstream/develop**
  - Si hay conflictos resolverlos
  - Si está todo OK, entonces mi **rama local "registrar-usuario"** debería estar por arriba de upstream/master (porque ya integre los cambios que había en upstream/master pero aún no subí los cambios de mi rama, es decir, mi rama tiene más cambios)

5) Hacer Push (con esto subo los cambios de mi rama local **registrar-usuario**, a mi **rama remota**, que sería **origin/registrar-usuario**). Como la rama es nueva (fue creada en el paso 2) cuando hago el push me va a preguntar:



Hay que darle que [Sí]

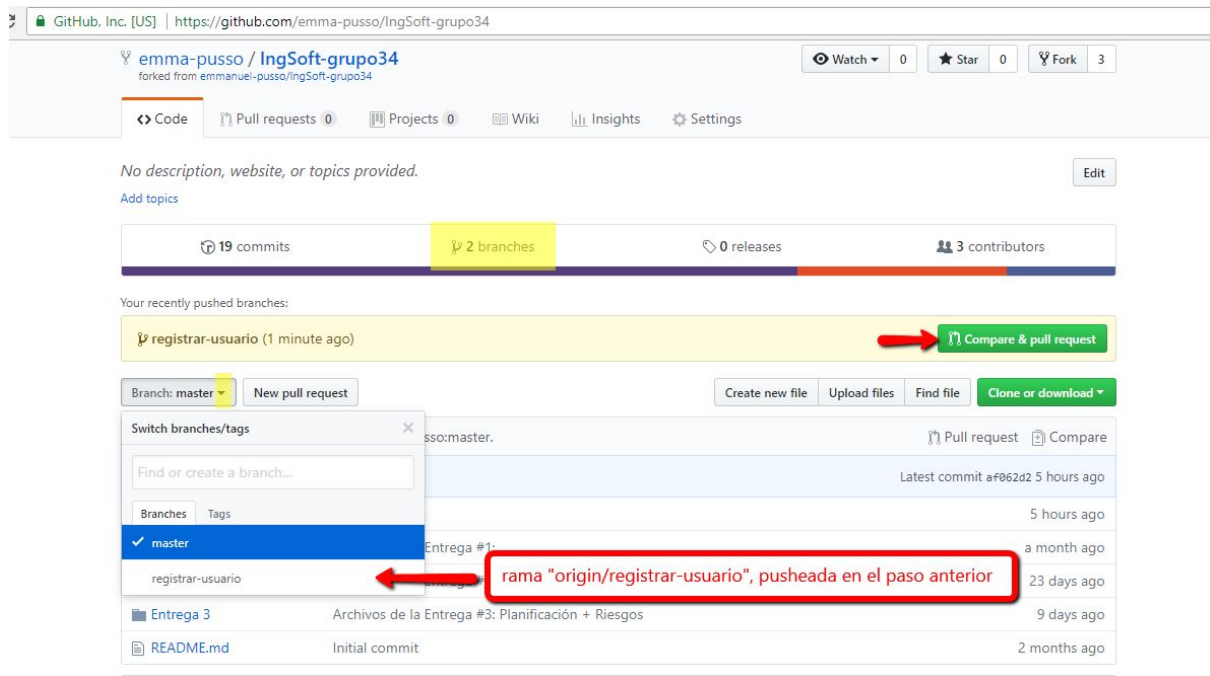


Hay que darle que [Sí]

Luego de ello se creo la rama remota **origin/registrar-usuario**, y se pushearon mis cambios de local a esa rama.

6) En Github, **acceder al repo remoto origin (fork)**, que sería x ejemplo:

<https://github.com/matias-ocares/IngSoft-grupo34>

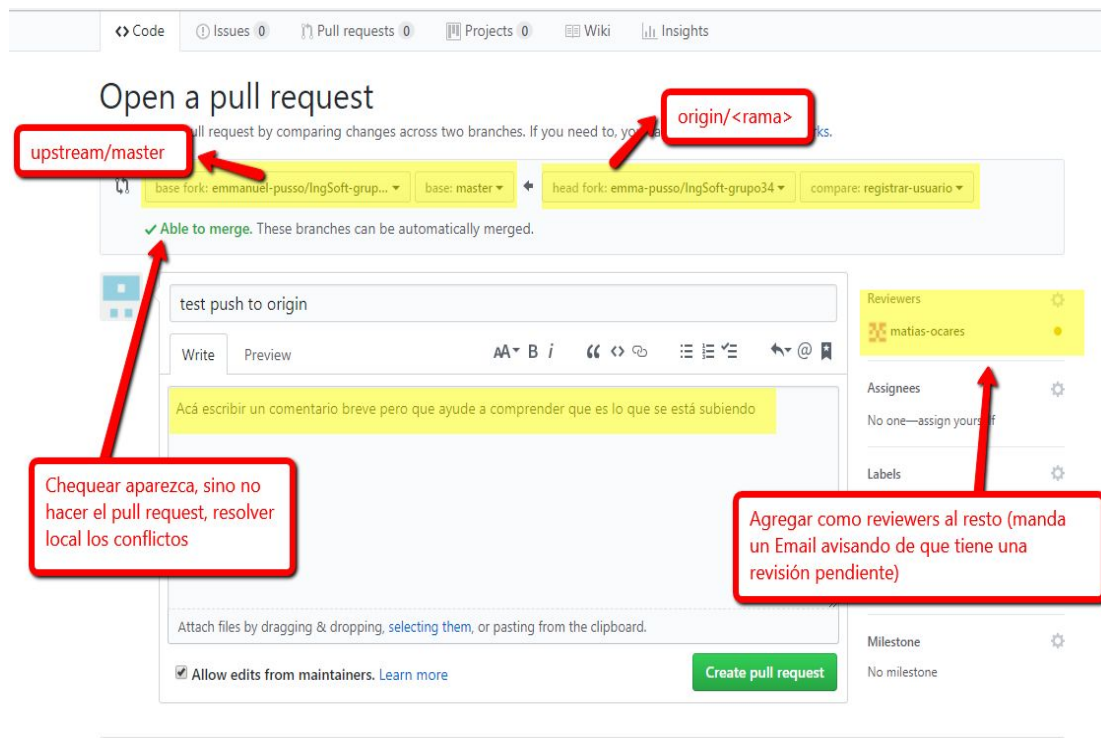


Pueden ver que se creó la rama remota **origin/registrar-usuario**, y aparece el botón **[Compare & pull request]** que es a partir de donde van a hacer el pull request.

#### 7) Click en [Compare & pull request]

- Verificar que aparezca el texto “Able to merge”
- Agregar una descripción, y seleccionar al resto como reviewers (les va a llegar un Email con el pedido de code review)
- Verificar que en **base fork** aparezca el **repo upstream** “**emmanuel-pusso/lngSoft-grupo34.git**” y la rama “**master**”
- Verificar que en **head fork** aparezca el **repo origin** por ejemplo “**matias-ocares/lngSoft-grupo34.git**” y la rama en la que hayan estado trabajando, para el ejemplo sería “**registrar-usuario**”

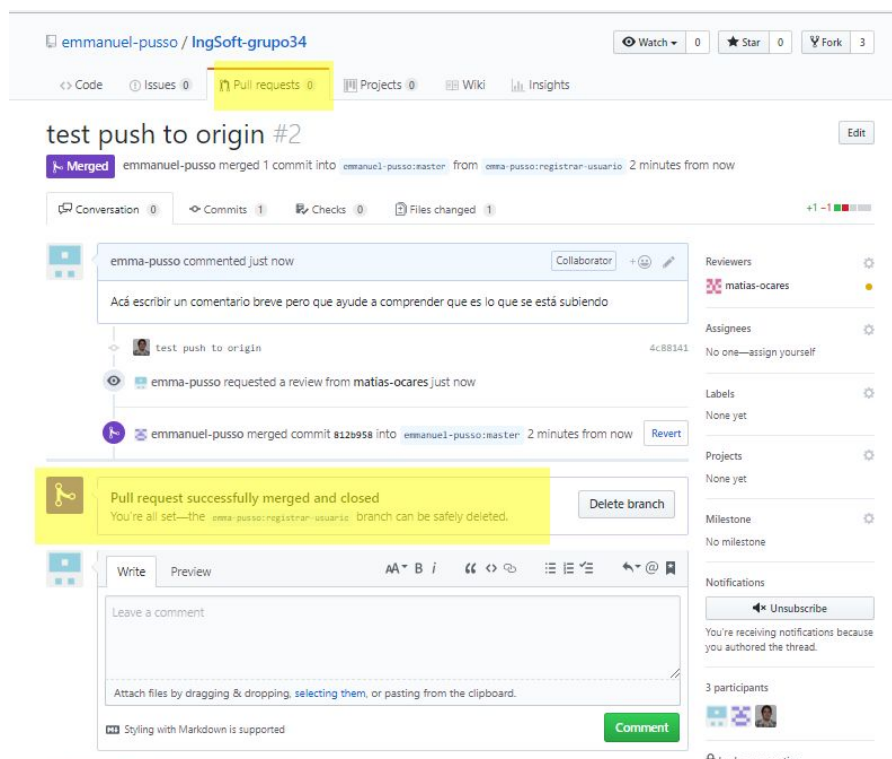




Click en [Create pull request]

Ese pull request lo recibo yo (que soy el creador del repositorio), y una vez que lo apruebo se integran los cambios con upstream/master, y recién ahí están disponibles para TODOS.

Luego de ello lo van a ver como **"Pull request succesfully merged and closed"**





8) Mantener la rama hasta que se hayan integrado los cambios, después pueden borrarla. Cuando vaya a desarrollar una nueva HU, repetir los pasos desde el principio (creando una nueva rama).