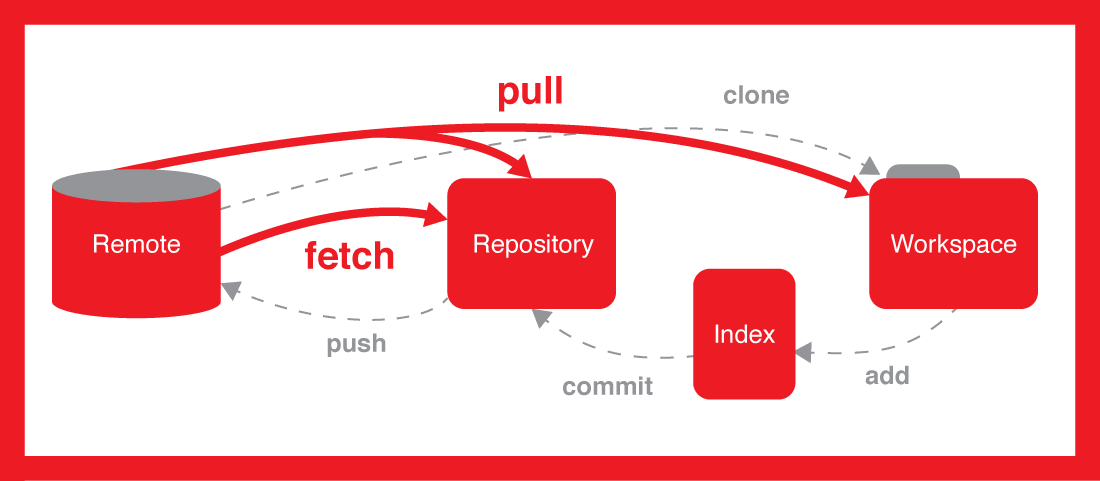
[](https://www.redeszone.net/2013/06/08/utilizar-git-y-github-para-subir-proyectos-desde-ubuntu-ii/)[](https://www.google.com.ar/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjCj-nij73dAhXCDZAKHW9jAHgQjRx6BAgBEAU&url=https://www.exa.unicen.edu.ar/es/estudios/catedras&psig=AOvVaw0Y1gyKb3KHZF6Wkn00uGln&ust=1537104804117718)

trabajo especial

Git – Github

Tecnología de la información en las organizaciones | 25 de septiembre de 2018

Grupo: PINEL, Federico; HSIEH, Ezequiel; HSIEH, María Teresa

INFORME GITHUB

1. **Para comenzar a usar GIT**

Es indispensable tener la aplicación git instalada en la computadora y una cuenta creada en github.com para poder comenzar a trabajar con esta herramienta, una vez ya creada iremos a la aplicación llamada git bush y allí colocaremos el primer comando que sirve para introducir el nombre de usuario en la aplicación y el email.

Comando nombre de usuario: *git config --global user.name  “usuario”*

Comando E-mail*: git config --global user.email* [*email@ejemplo.com*](mailto:email@ejemplo.com)

Al concluir estos pasos se deberá crear una carpeta que será la que va a contener todos los repositorios que nosotros coloquemos en este caso la carpeta llevará el nombre de “Repositorios” que estará en el escritorio.

Comando para ubicarnos en el escritorio: *cd desktop*

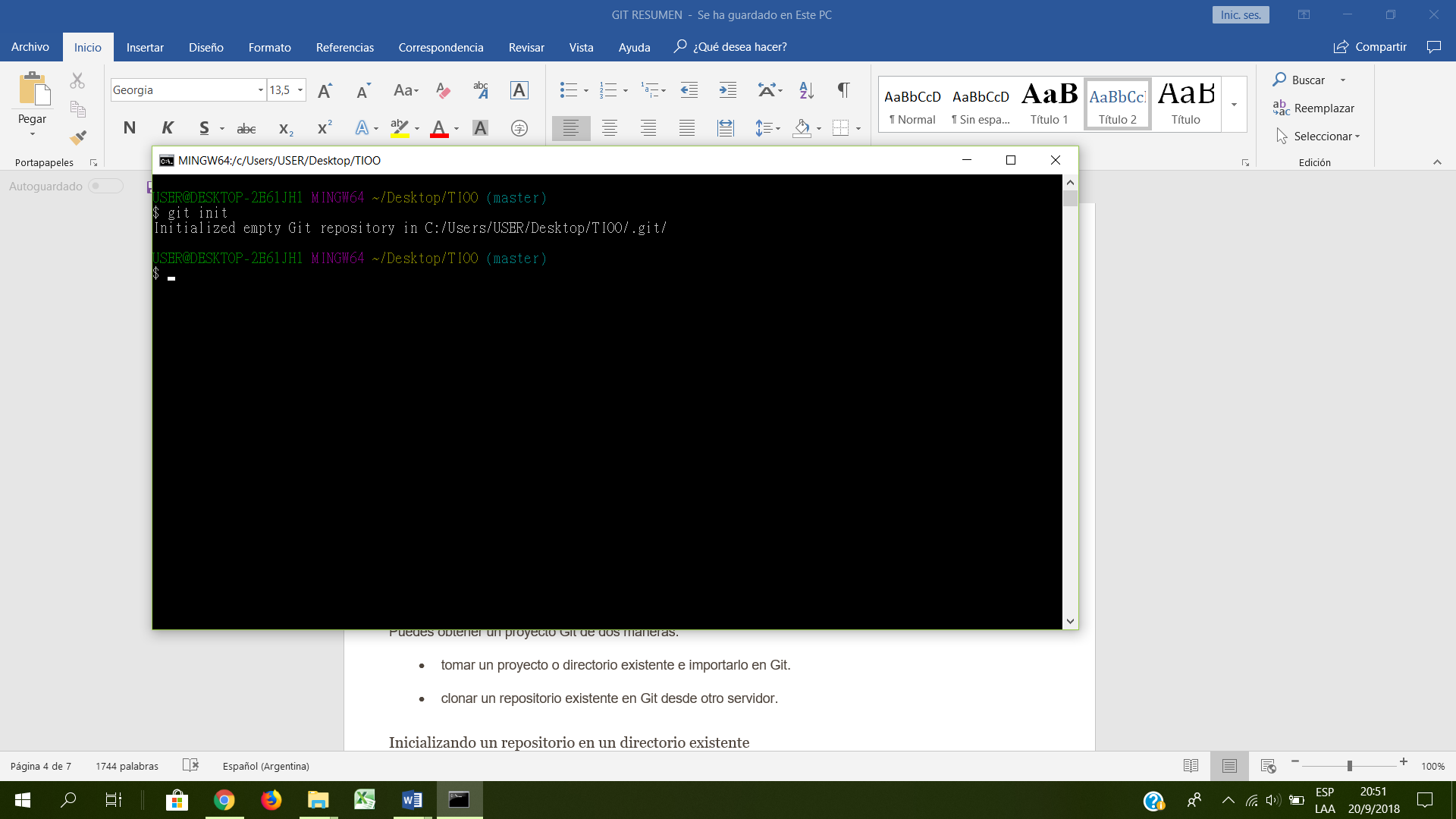
Comando para ubicarnos dentro de la carpeta: *cd Repositorios*

1. **Obteniendo un repositorio Git**

Puedes obtener un proyecto Git de dos maneras:

1. tomar un proyecto o directorio existente e importarlo en Git.
2. clonar un repositorio existente en Git desde otro servidor.

### **Inicializando un repositorio en un directorio existente**

Si estás empezando a seguir un proyecto existente en Git, debes ir al directorio del proyecto y usar el siguiente comando:  *git init*

Esto crea un subdirectorio nuevo llamado .git, el cual contiene todos los archivos necesarios del repositorio. Todavía no hay nada en tu proyecto que esté bajo seguimiento.

Luego ponemos: *git add* ***nombre del archivo***

Para especificar qué archivos quieres controlar o colocar el proyecto dentro de la carpeta utilizamos el comando:

*git add* ***nombre del archivo***, seguidos de un *git commit* para confirmar los cambios:

Al concluir esto deberemos corroborar el estado del proyecto agregado utilizando el comando *git status,* este comando te permite conocer en qué estado se encuentra el proyecto y en que rama(branch) se encuentra. *Ramificaciones (véase punto3*)

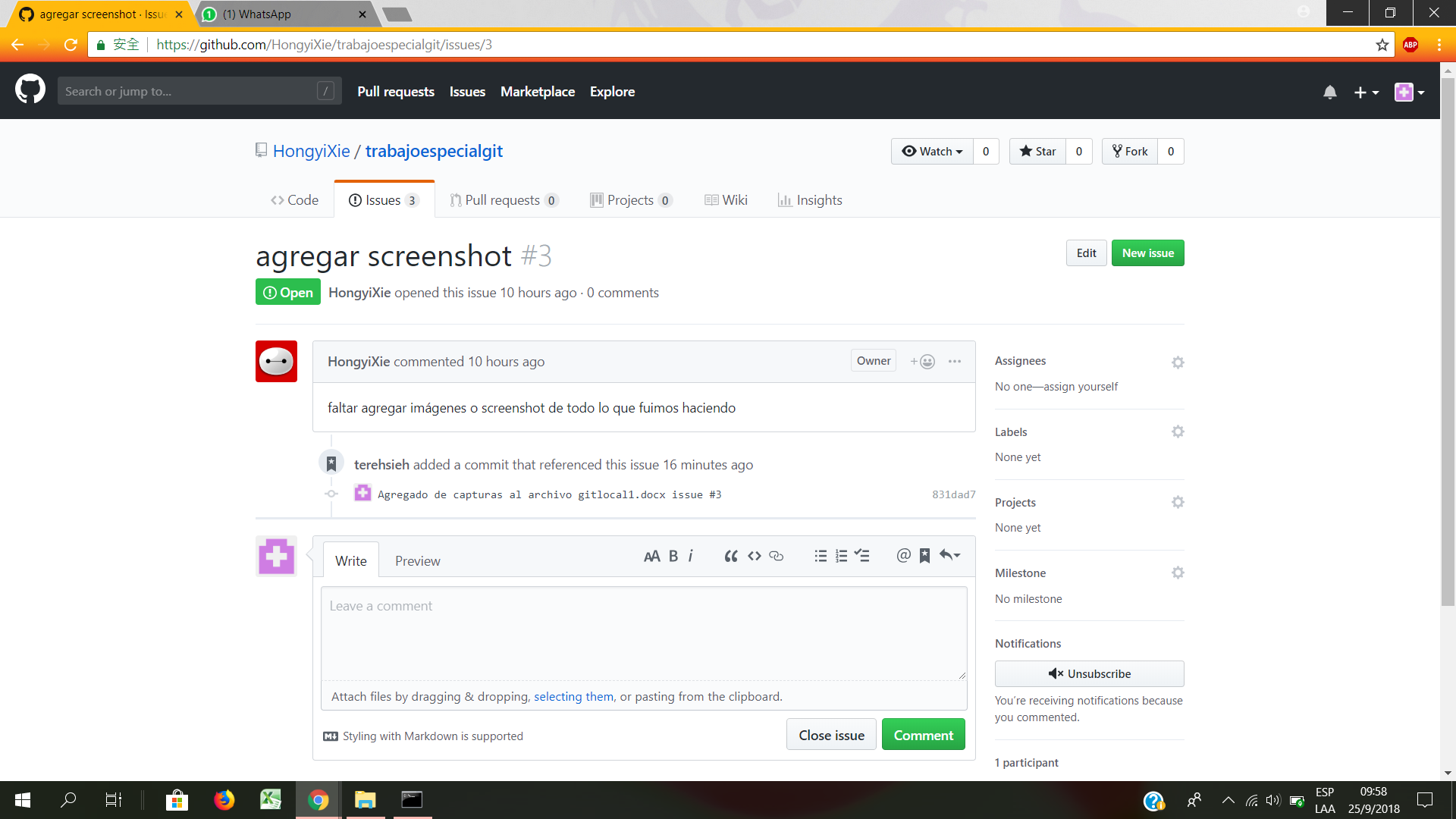
Una vez finalizado esto se deberá hacer un commit, que se refiere al mensaje de confirmación para este proyecto con el comando:

*git commit -m “****mensaje****”*

Dentro del mensaje también se puede enlazar el issue con #**n°del issue** *(véase punto 4)* **que se quiere enlazar**

Ejemplo: git commit -m “Agregado de capturas al archivo gitlocal1.docx issue #3”

Al subir los cambios me enlaza automáticamente con el issue#3 (agregar screenshot #3)



Al estar con el commit colocado el trabajo ya estará disponible para subirlo a la página github.

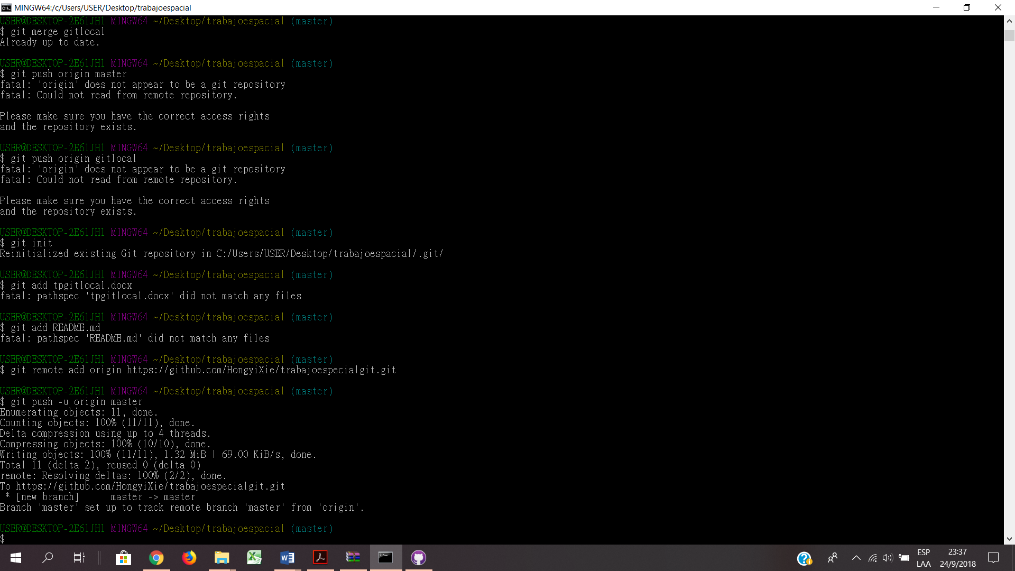
**Subir los cambios realizados**

Para subir por primera vez los cambios es necesario configurar el lugar donde se van a subir los cambios.

*git remote add origin* ***link obtenido de las instrucción de github***

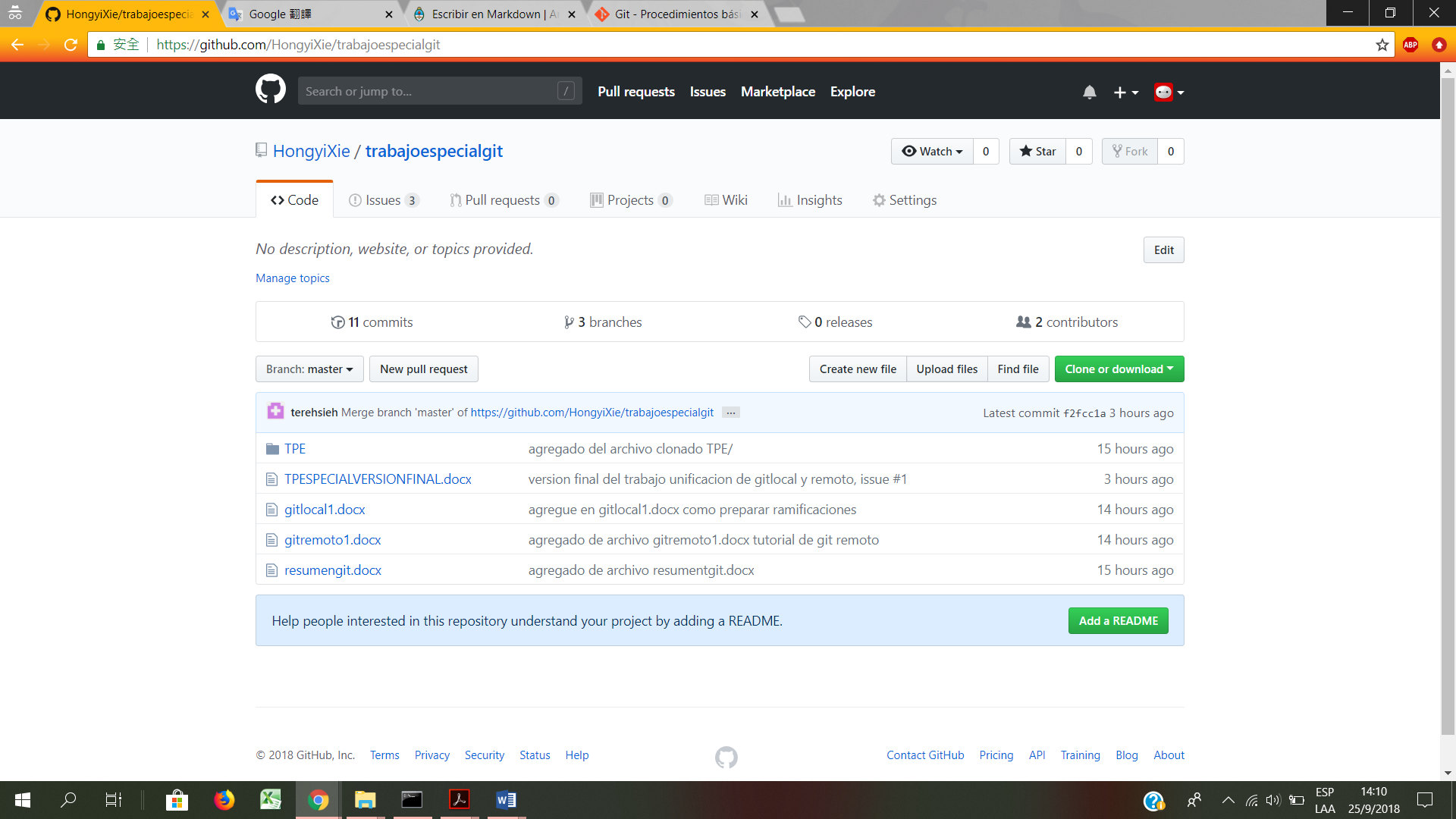
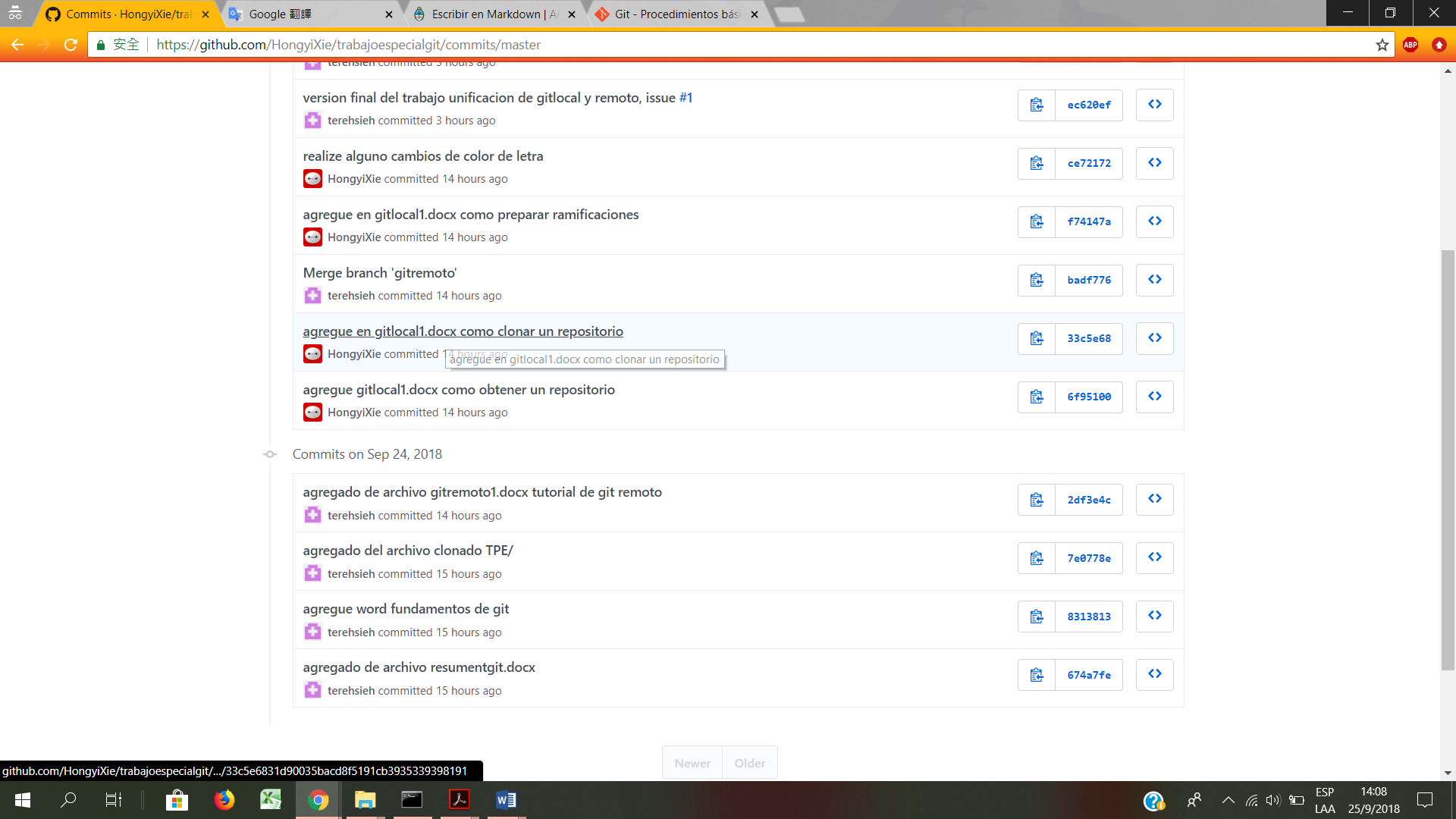
Una vez configurado, cada vez que se quiera subir los cambios solo es necesario utilizar el siguiente comando.

*git push -u origin* ***nombredelarama***



Al hacer esto nos pedirá nombre de usuario y contraseña para poder habilitar al Git a subir ese archivo al repositorio de GitHub.

Al subir los cambios, podemos ver en GitHub todos los cambios o archivos que subimos.



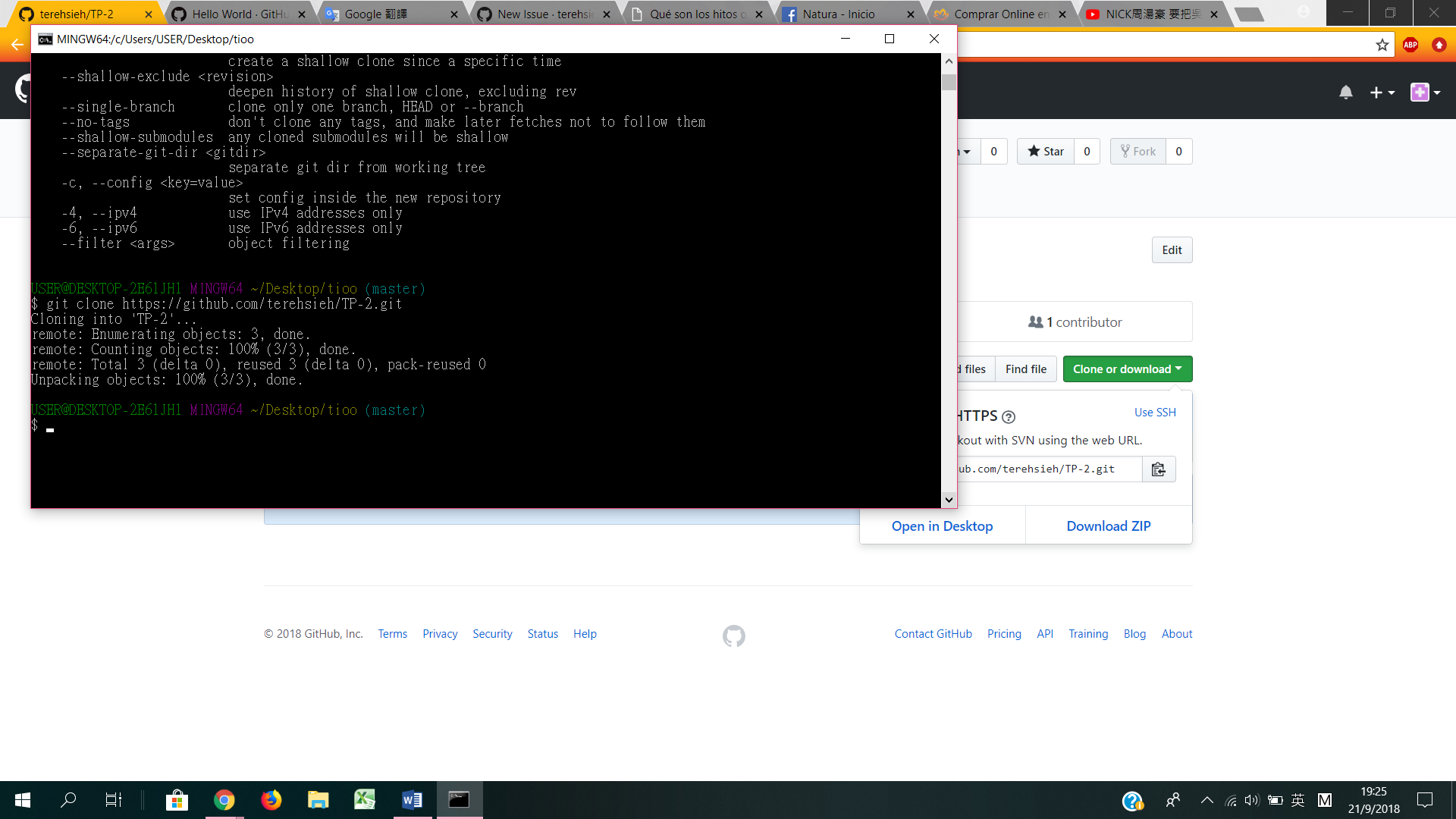


Para fusionar todos los cambios que se han hecho en el repositorio local trabajando, el comando que se usa es *git pull.*

### **Clonando un repositorio existente**

Si deseas obtener una copia de un repositorio Git existente — por ejemplo, un proyecto en el que te gustaría contribuir — el comando que necesitas es *git clone [url]*

Ej: $ git clone https://github.com/terehsieh/TP-2.git



1. **Ramificaciones**

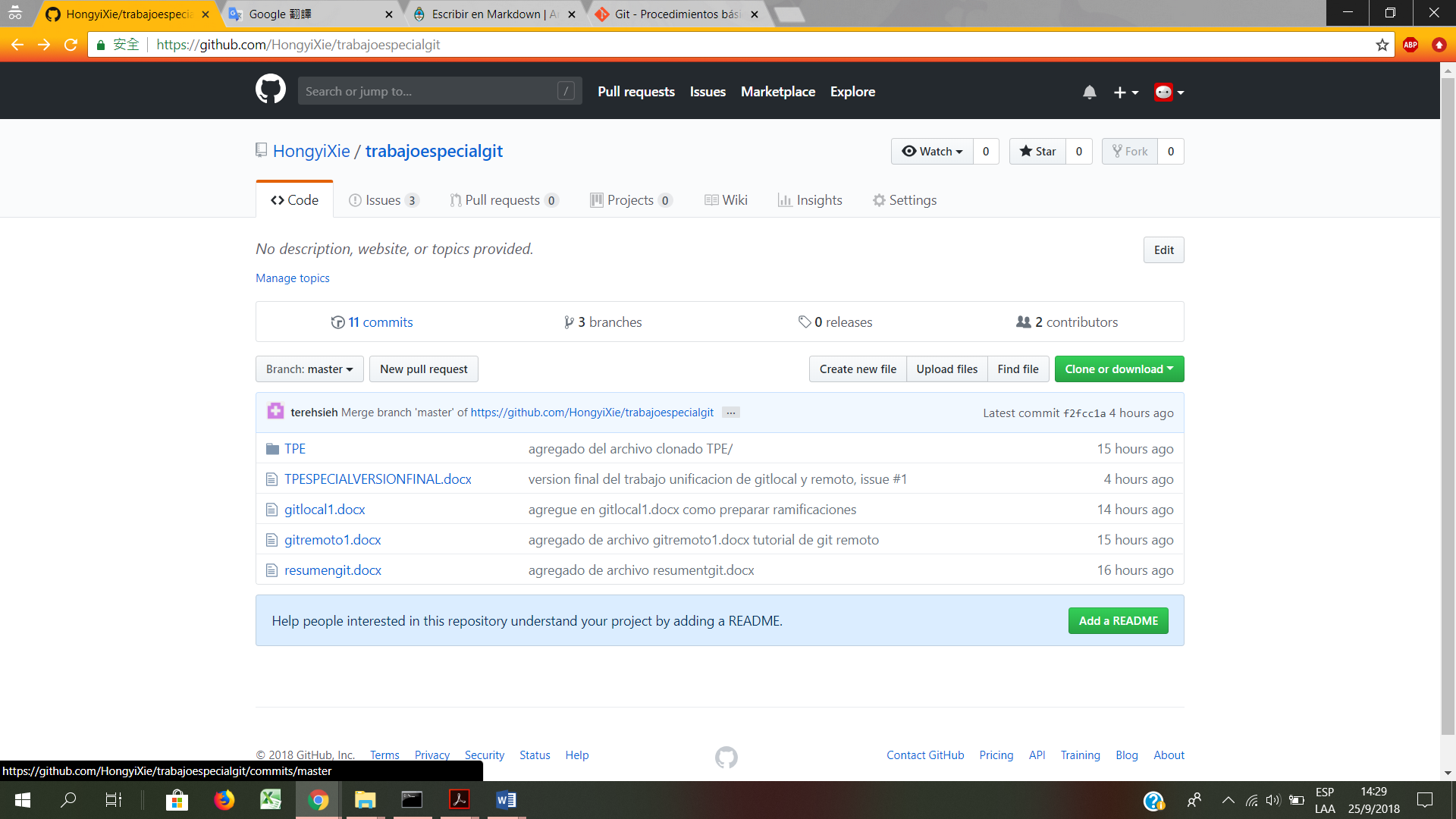
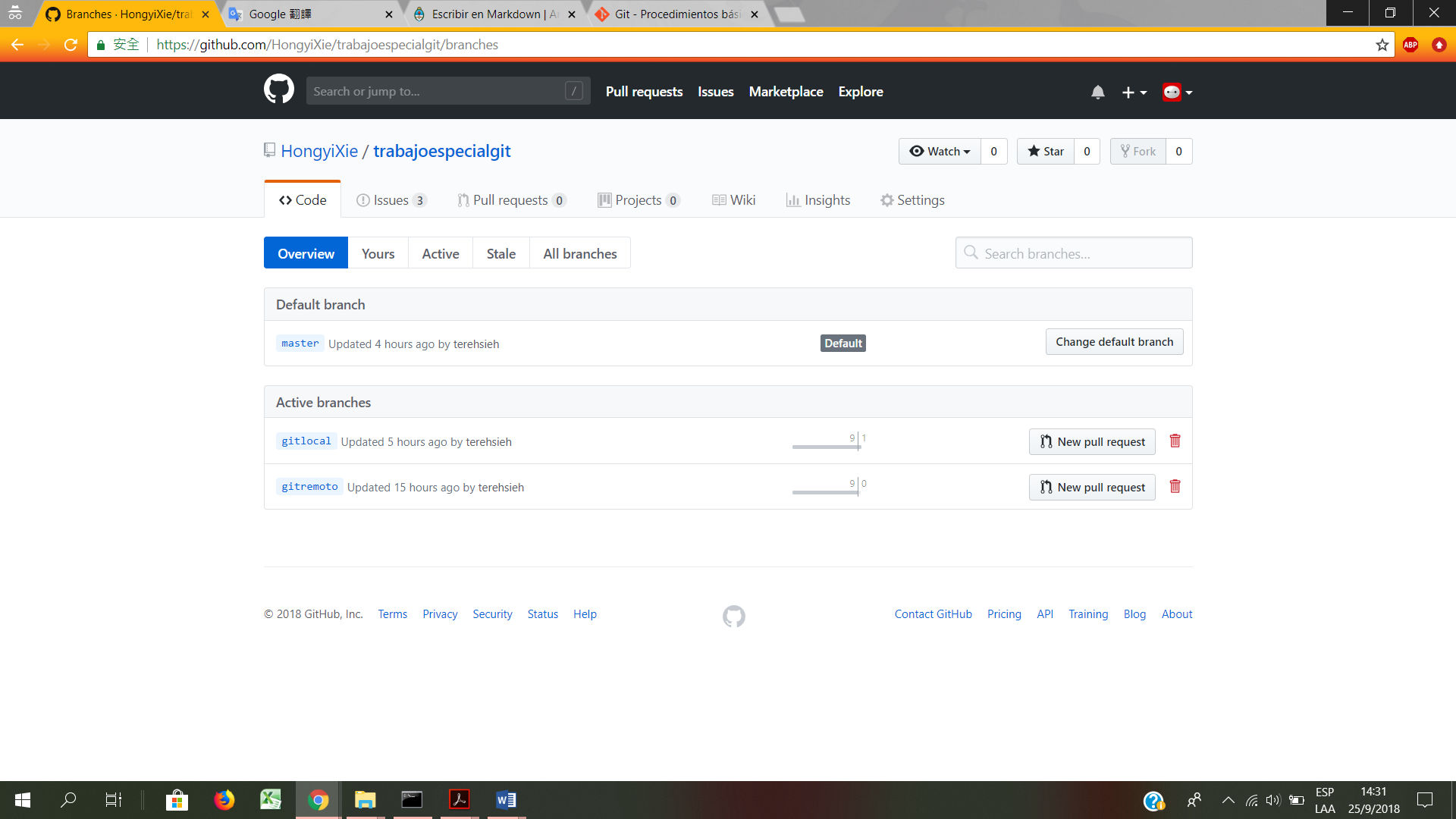
### **Crear una Rama Nueva**

Para crear una rama: *git branch* ***nombre de la rama***

Ej: *git branch gitlocal*

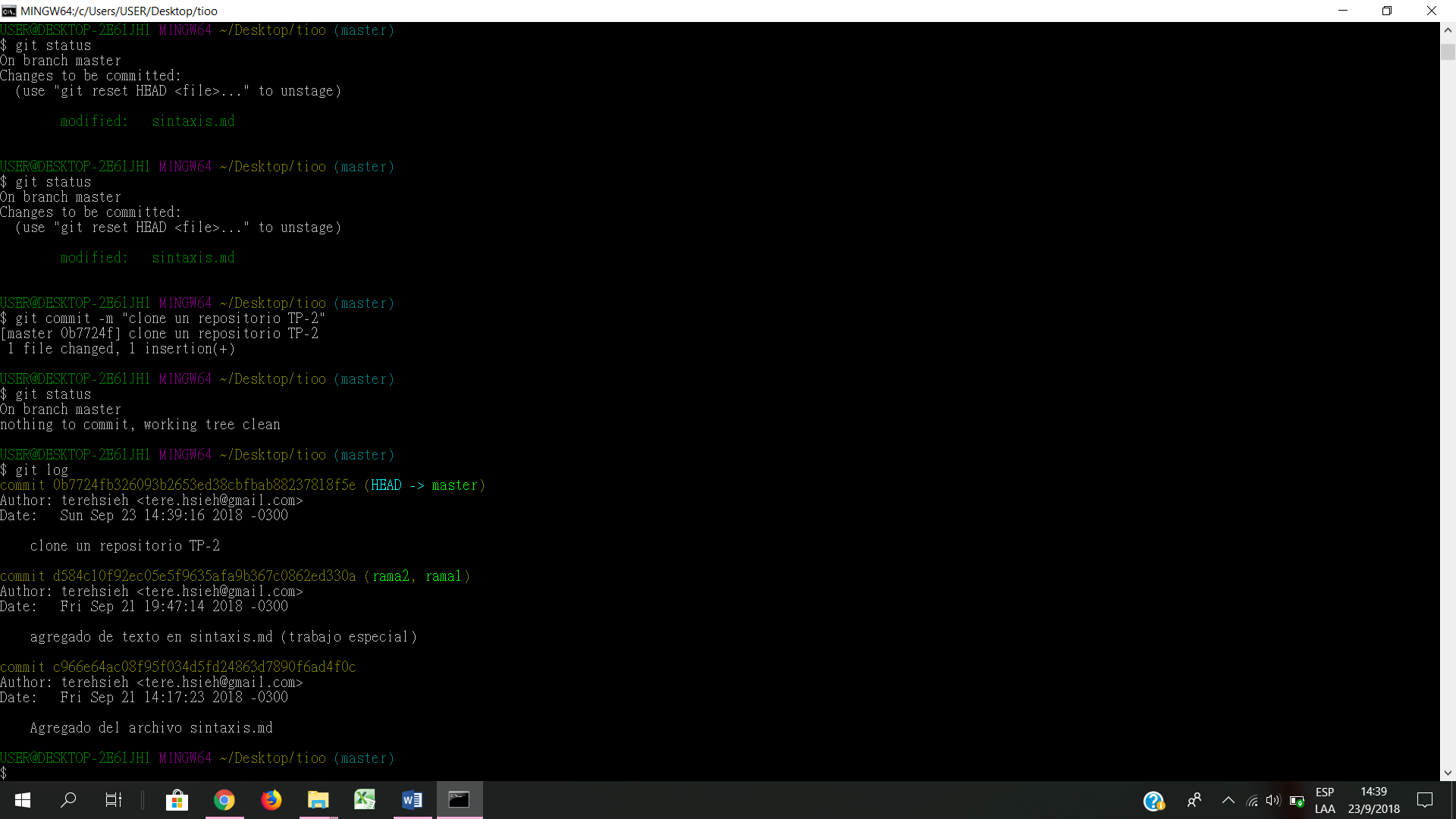
Eliminar una rama: *git branch -d* ***nombre de la rama***

Luego de trabajar en ella es necesario confirmar los cambios (*commit*) y volverlos a subir (*push*) o si algún otro colaborador ha realizado cambios podemos actualizarlos (*git pull)*





Obsérvese creación de 2 ramas (gitlocal/gitremoto)

**Cambio de ramificación**:

Para saltar de una rama a otra, tienes que utilizar el comando:

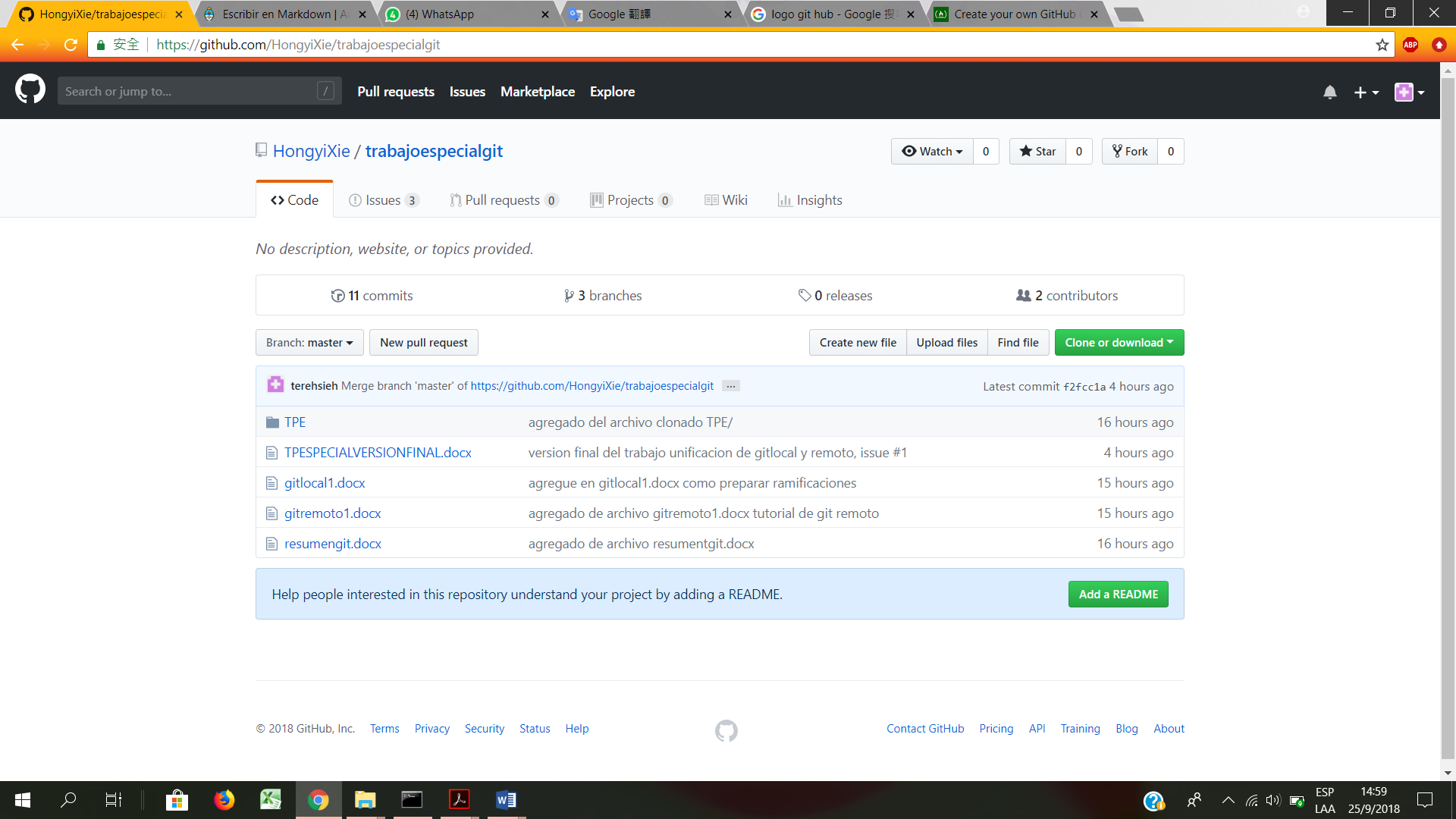
*git checkout* ***nombre de la rama***.

Con *git log*: proporciona una lista cronológica de cambios realizados detalladamente sobre el proyecto; e indica en qué rama está trabajando, con el apuntador HEAD🡪 **nombre de la rama.**

**Unir ramas:**

El comando *git merge* ***nombre de la rama:*** incorpora los cambios a la rama master para ponerlos en producción.

Una vez subidos los cambios es aparece en Github el *merge.*



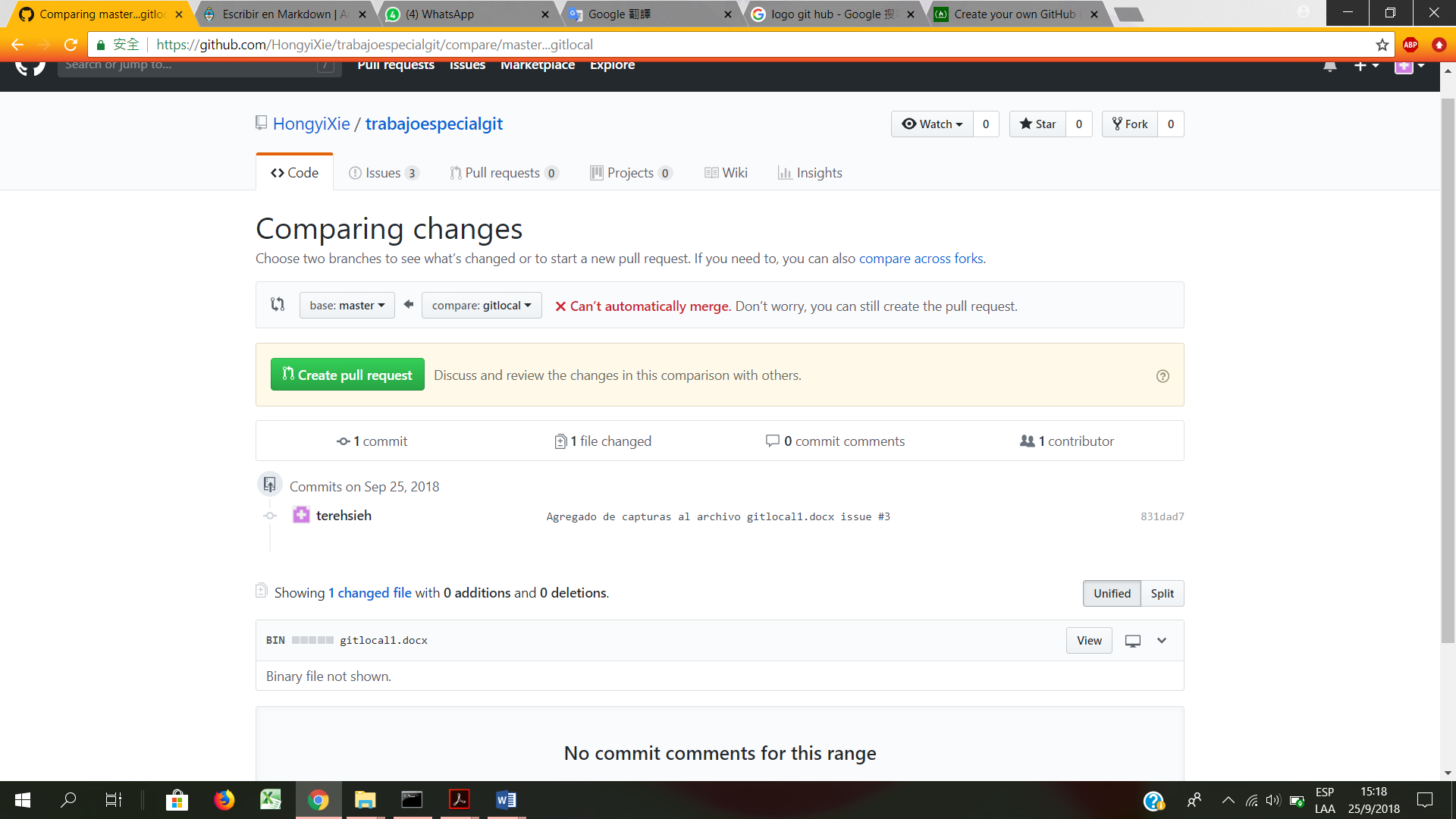


Al unir las ramas puede ser que surjan conflictos

Comando *git diff --base* <**nombre archivo**>: para ver conflictos entre archivos.

Comando *git diff <source-branch> <target-branch*>: para ver conflictos entre distintas ramas.

O puedes seleccionar el Github donde dice branches, clickear en el branch que quieras comparar y ver las diferencias.

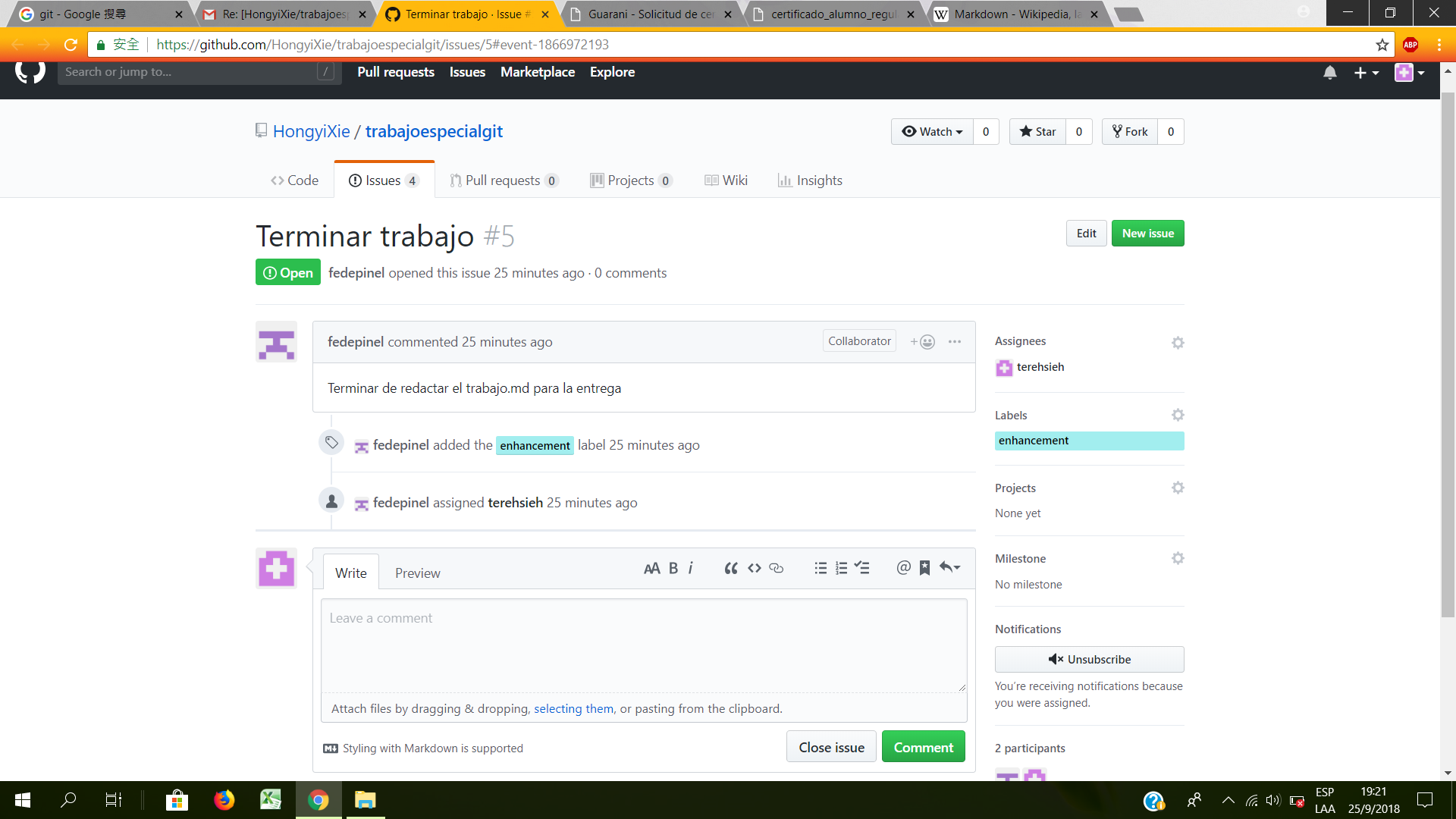


1. **Issues**

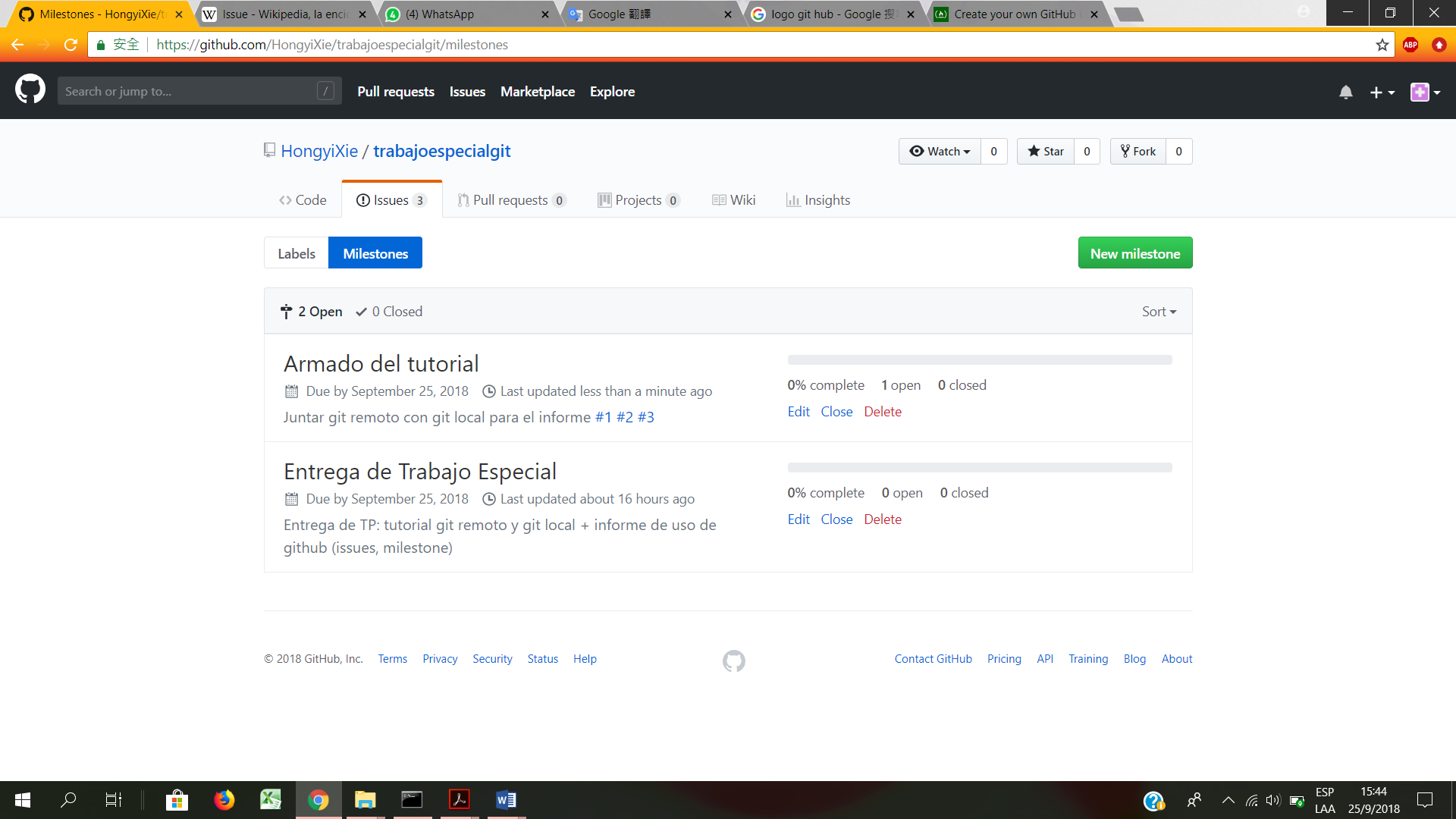
Issue es una unidad de trabajo para realizar una mejora en un Sistema informático. Un issue puede ser el arreglo de un fallo, una característica pedida, una tarea, un pedido de Documentación específico y todo tipo de solicitud al equipo de desarrollo.

Función: crear, actualizar y resolver incidentes reportados por usuarios o por otros empleados.

Para crear un issue tienes que ir a la página de GitHub, clickear Issues- New issue.

Componentes de una tarea:

* Titulo
* Comentario: puedes usar Markdown *(véase punto 5*)
* Etiquetas(*labels*)
* Asignar un responsable (assignee)
* Estados (open/close)
* Hito(*milestone*):
  + Señala fecha de inicio y finalización del proyecto
  + Enlazar issues #n°issue



1. **Escribir en Markdown**

Guía básica para escribir en Markdown

**Negrita o itálica**

Escribir en negrita: **\*\*texto\*\***

Escribir en itálica: *\*texto\**

**Títulos**

Para que aparezca como títulos usamos: #titulo

Podes usar uno o más numerales para establecer el nivel de título, en orden ascendente.

**Enlaces:**

[descripción de la pagina](url)

Ej: [logodegithub]( <https://www.google.com.ar/search?ei=9juqW4iRMpCwwgSKvrrIDg&tbm=isch&q=git+logo&oq=&gs_l=#imgrc=JD5IM5zC63wasM>:)

**Listas**

Para listar sin numeración se puede usar asterisco, signo de suma o guión corto (\*, + o -) como si se tratara de viñetas

- Ítem 1

- Ítem 2

Para listar con números, usarlos seguidos de un punto como viñetas

**Imágenes**

1. Subí la imagen en el apartado **Insertar imagen** que aparece dentro del tipo página, página de libro o noticia.
2. Completá los campos **Texto alternativo** (sirve para los lectores de pantalla, normalmente describe la tipografía) y **Título**.
3. Posicioná el cursor en el campo de texto donde quieras que vaya la imagen y hacé click en el botón **Insert**.

En preview changes es posible ver cómo queda ya en el formato de títulos, con negrita, cursiva, listado con viñetas.

Una ves terminado es necesario guardar los cambios abajo del todo aparece el botón “commit changes”

