viernes, 19 de julio de 2019

Nombre: Rodrigo Garcia Castillo

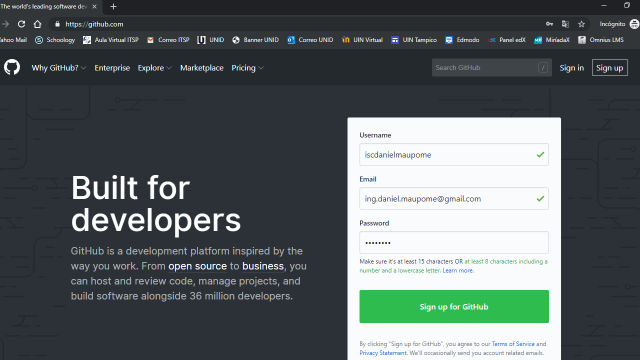
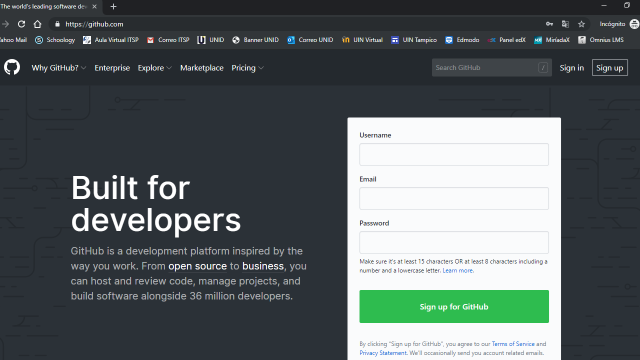
Daniel Arturo Maupomé Rosales

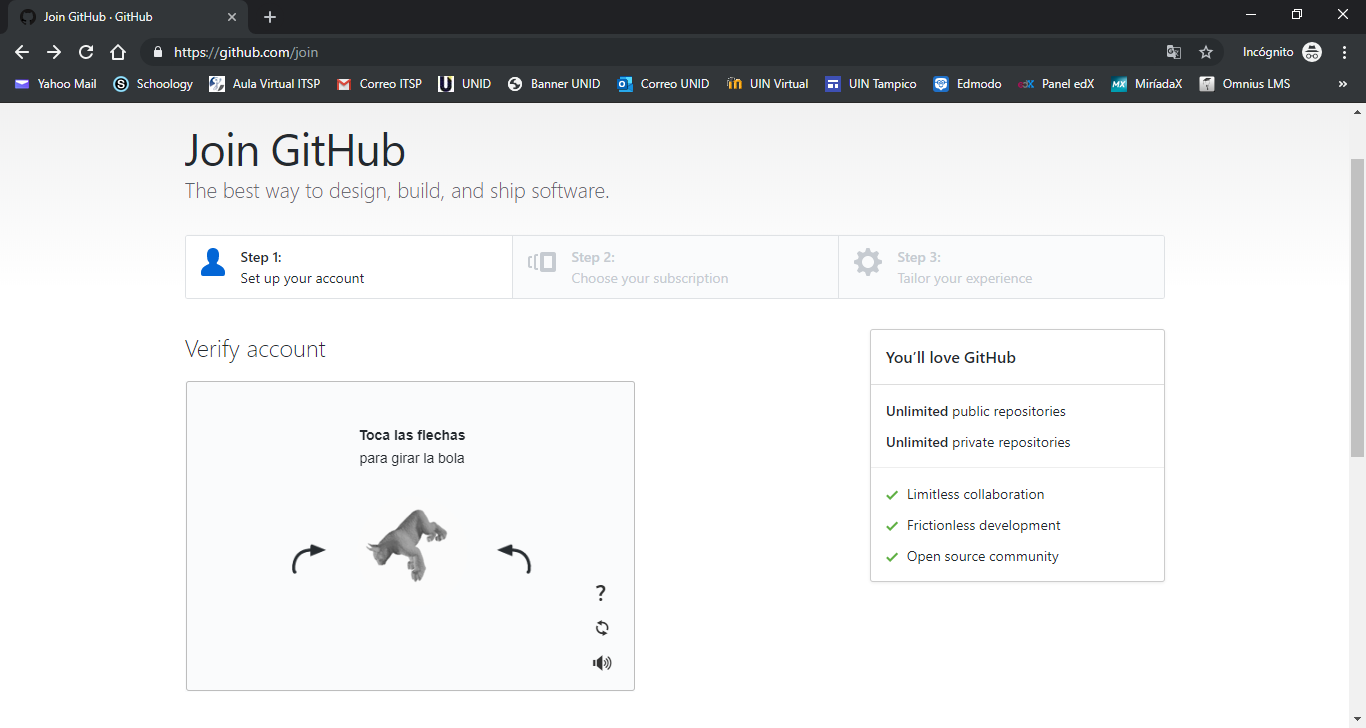
Juan Luis Olvera Ibarra

Jesus Cortes Ponce

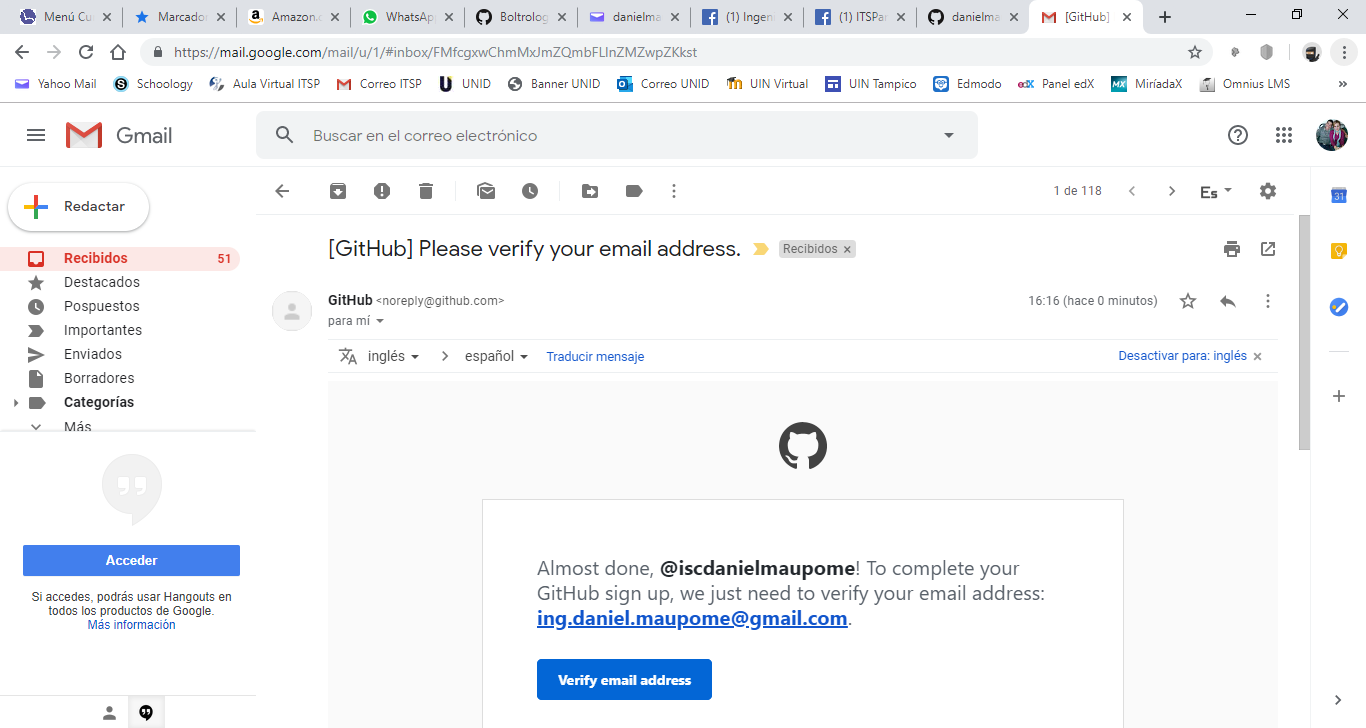
Manual de Git y GitHub.

Crear una Cuenta.

1. Ir a la página [www.github.com](http://www.github.com). Podemos ver la siguiente imagen en nuestra pantalla
2. Llenar los campos solicitados para crear nuestra cuenta gratuita, o si tienes cuenta, iniciar sesión. En este caso, se explicará el proceso de creación de una cuenta, tal como se aprecia en la imagen. Nota: No podemos usar puntos para el username, si quieres separar palabras, puedes usar el guion medio (-).
3. Para comprobar que eres humano, debes realizar un sencillo test que se describe en la siguiente imagen (básicamente, se debe enderezar una imagen)



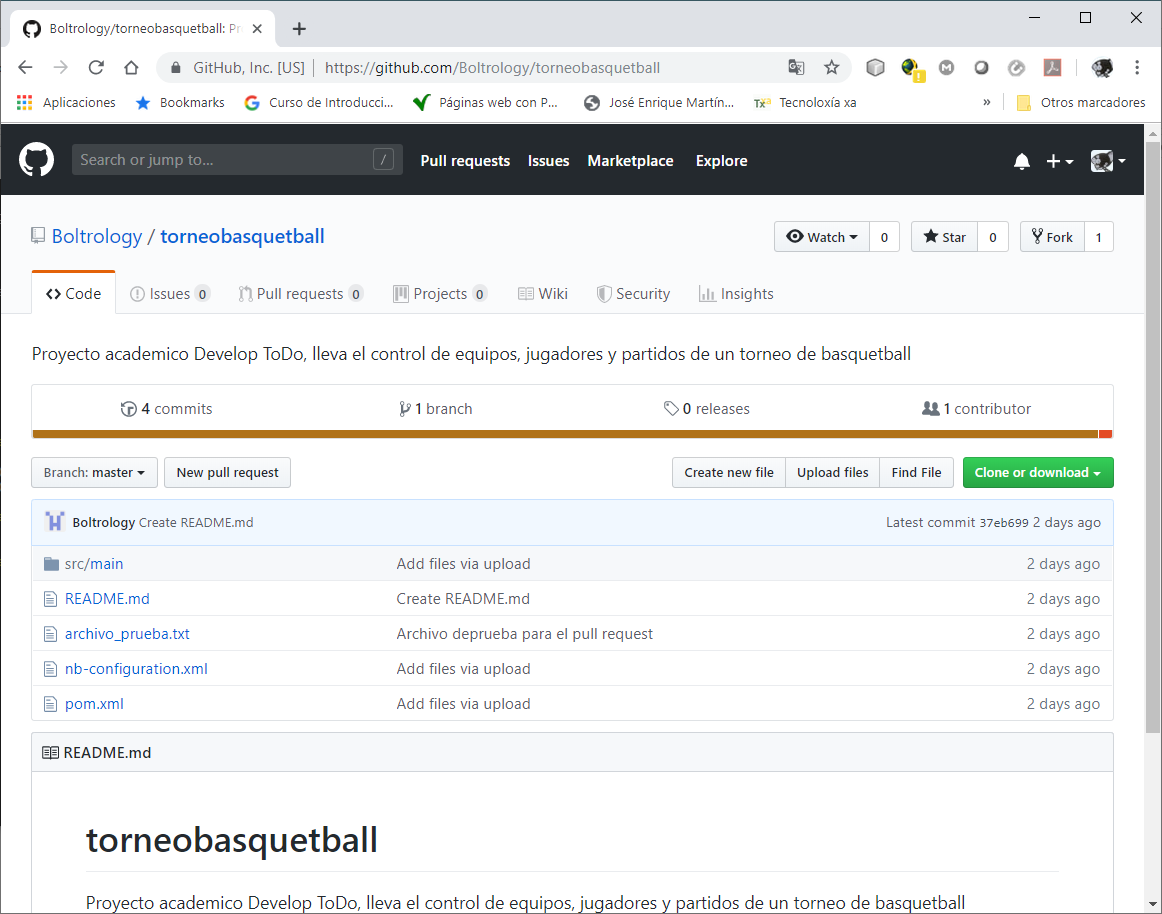
1. Posterior a elegir la suscripción, la cual es la free, te enviarán un enlace a tu correo para validar la cuenta, lo cual hay que hacer clic en el enlace que se envió a tu correo



1. ¡Listo! Tu cuenta en GitHub ha sido creada. Adicionalmente, puedes descargar GitHub Desktop, para tener un acceso de primera mano a tu repositorio desde tu equipo de cómputo. Más adelante veremos la instalación y/o configuración de los componentes necesarios desde NetBeans.

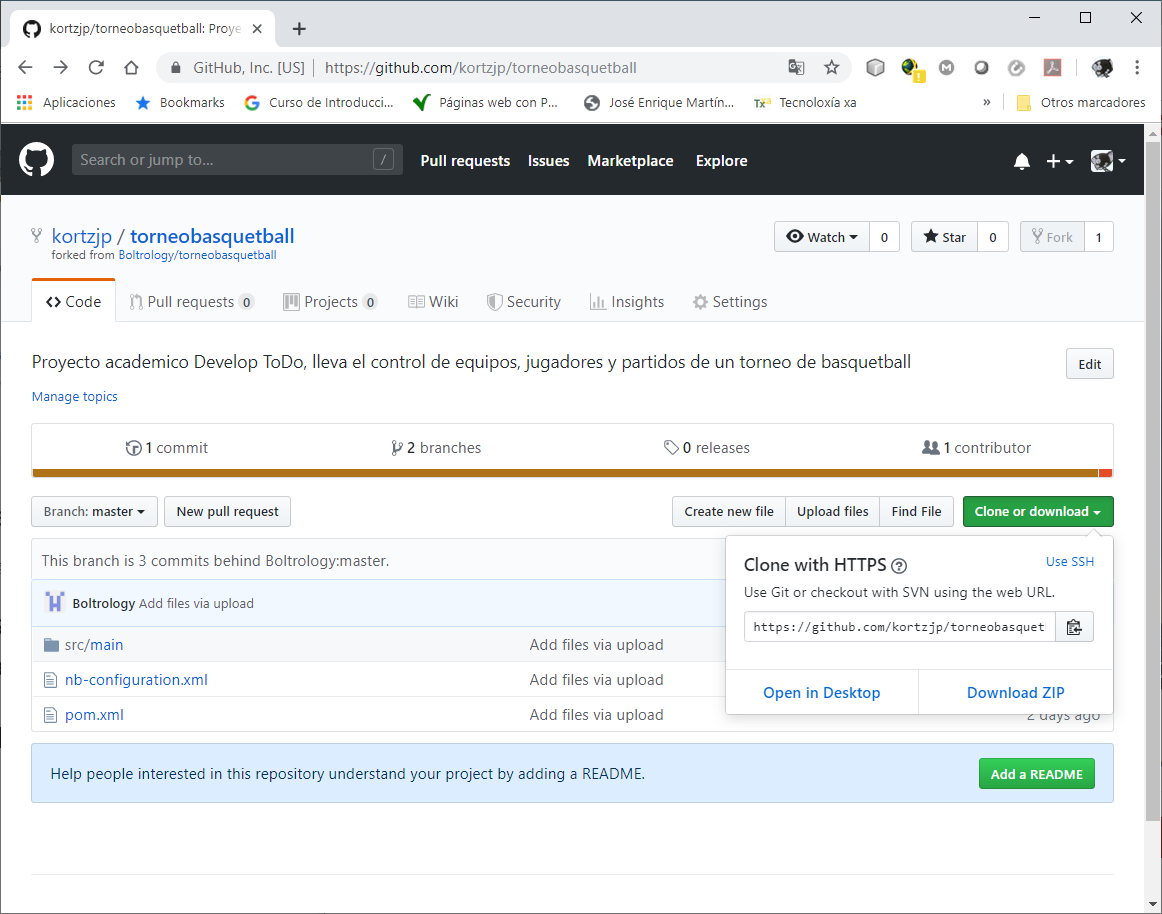
Subir un cambio vía web y NetBeans

1. Primero vamos a realizar un fork del repositorio origen, que por lo regular es de otro desarrollador(**Boltrology**).

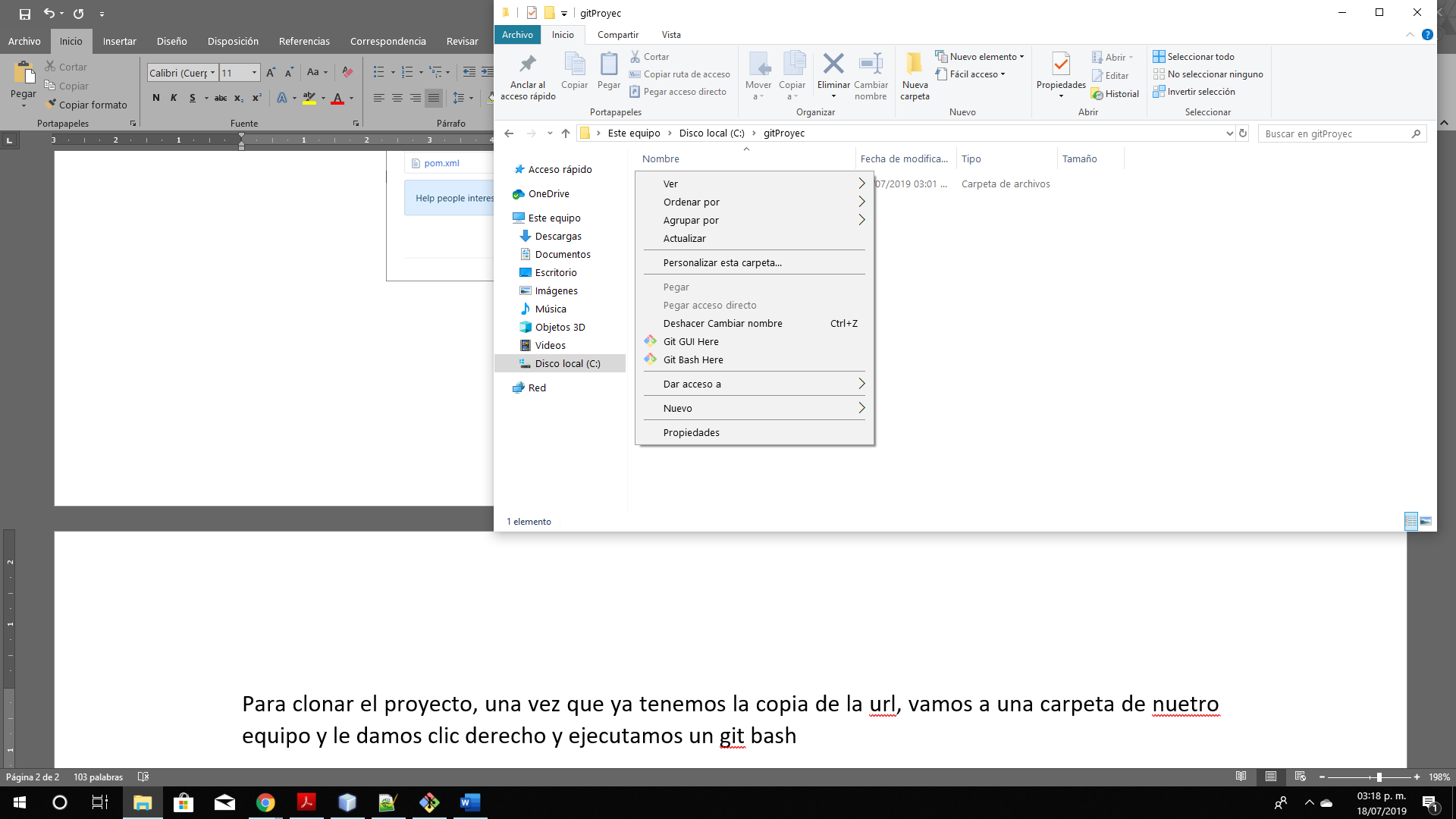


1. Un fork es una copia exacta de un repositorio original, con todo el historial hasta el momento de crear el fork. Esto lo hacemos para poder trabajar en el repositorio como si fuera nuestro(**kortzjp**).

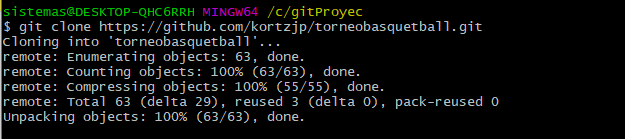
Ahora ya tenemos un repositorio el cual vamos a clonar a nuestra maquina local.



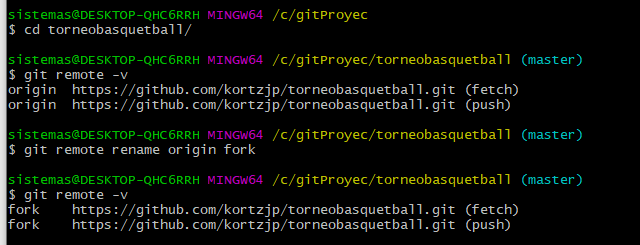
1. Para clonar el proyecto, una vez que ya tenemos la copia de la url, vamos a una carpeta de nuetro equipo y le damos clic derecho y lejimos un Git Bash Here para que se abra la consola de Git



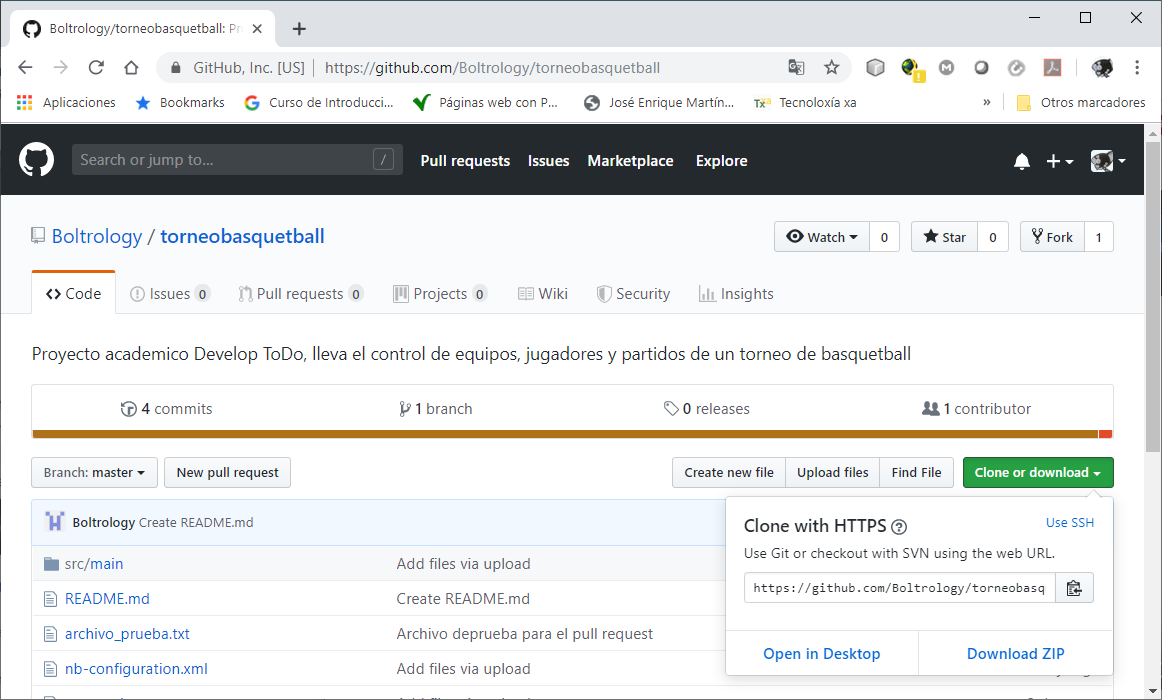
1. Estando en la consola solo colocamos el comando **git clone y la url del repositorio**, y damos enter para que se descargue el repositorio



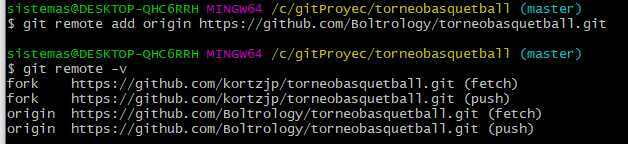
1. Con esto tendremos el repositorio en nuestra maquina local, ahora entramos al repositorio con el comando **cd** y como queremos mantener un enlace al repositorio fork(kortzjp) y al original(Boltrology) vamos a renombar al fork con el comando **git remote rename origin fork** y verificamos el cambio con el comando **git remote -v**



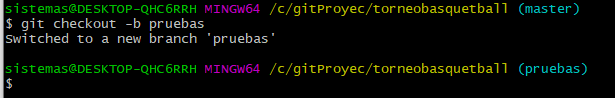
1. Ahora vamos a enlazar el repositorio original de **Boltrology,** para esto vamos al repositorio original y copiamos la url de la opción de **clone or download**



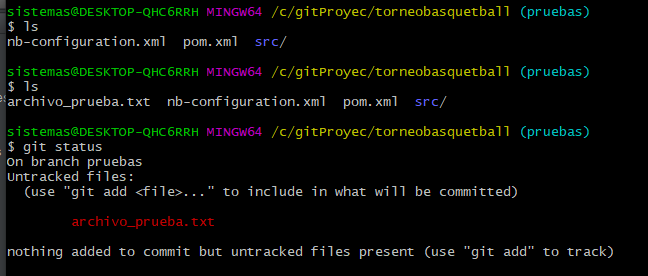
1. Ahora regresamos a la consola de git, con el comando **git remote add origin y la url** del repositorio original enlazamos el repositorio. El nombre de **origin** puede ser cualquiera, pero se decidió ocupar origin para tener en claro que pertenece a **Boltrology** y el fork es el de **kortzjp.** Verificamos los cambios con el comando **git remote -v**



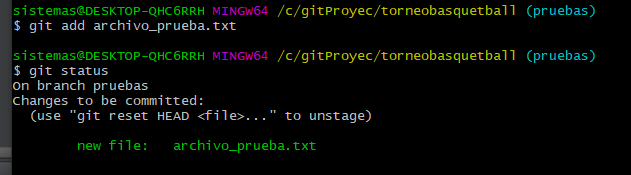
1. Con esto ya tenemos preparado todo para comenzar a realizar modificaciones o contribuciones; pero se recomienda que no se hagan en la rama **master**, para esto vamos a crear otra rama con el comando **git checkout -b pruebas**, el nombre pruebas lo puedes cambiar por el que gustes. Después de ejecutar el comando también nos hemos movido a la nueva rama.

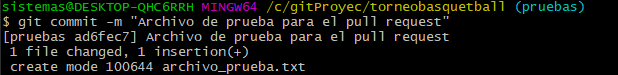


1. Ahora podemos modificar algo del proyecto, para este caso se ha creado **archivo\_prueba.txt**  y con el comando **git status** se ha detectado el cambio

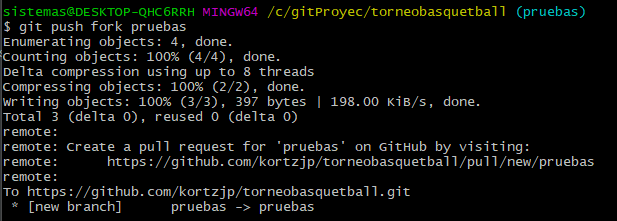


1. Ahora con el comando **git add archivo\_prueba.txt** se prepara el archivo para después realizar el commit con su mensaje correspondiente

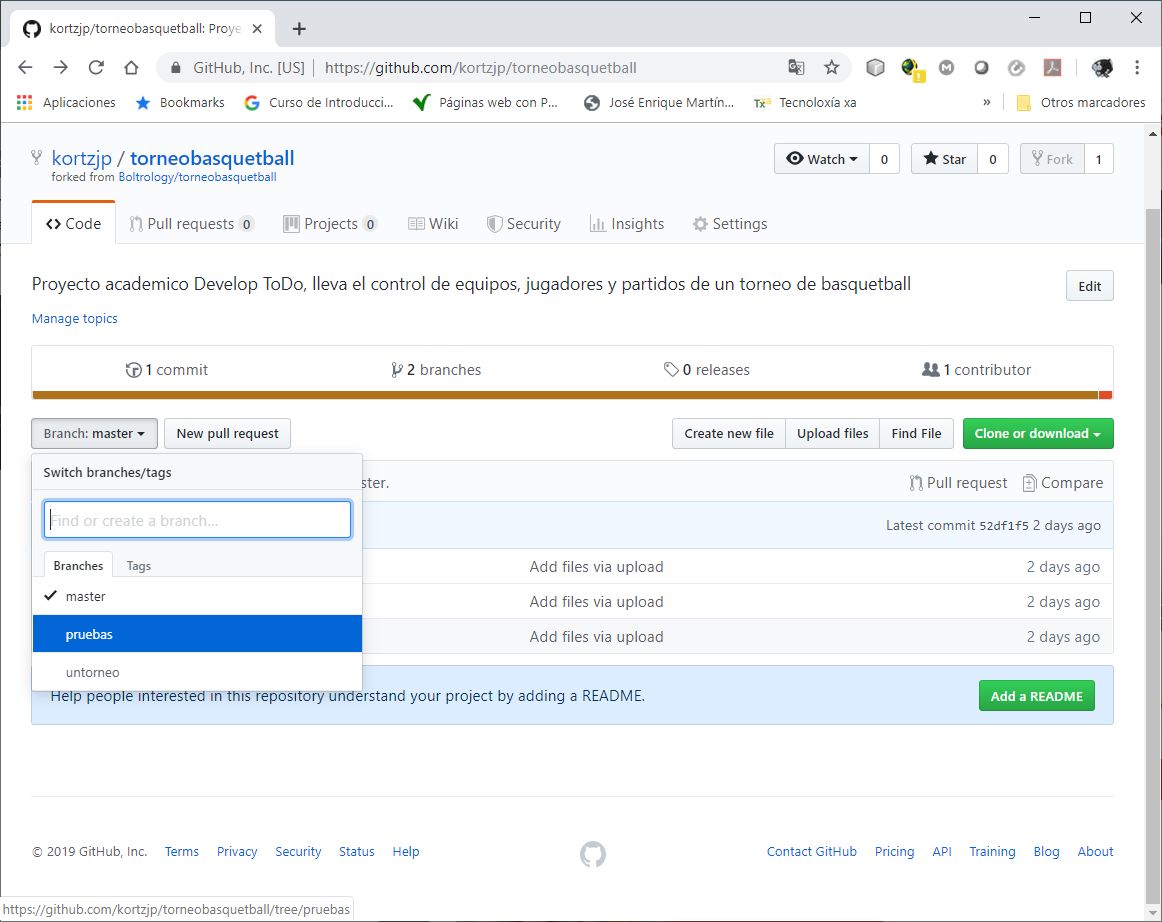




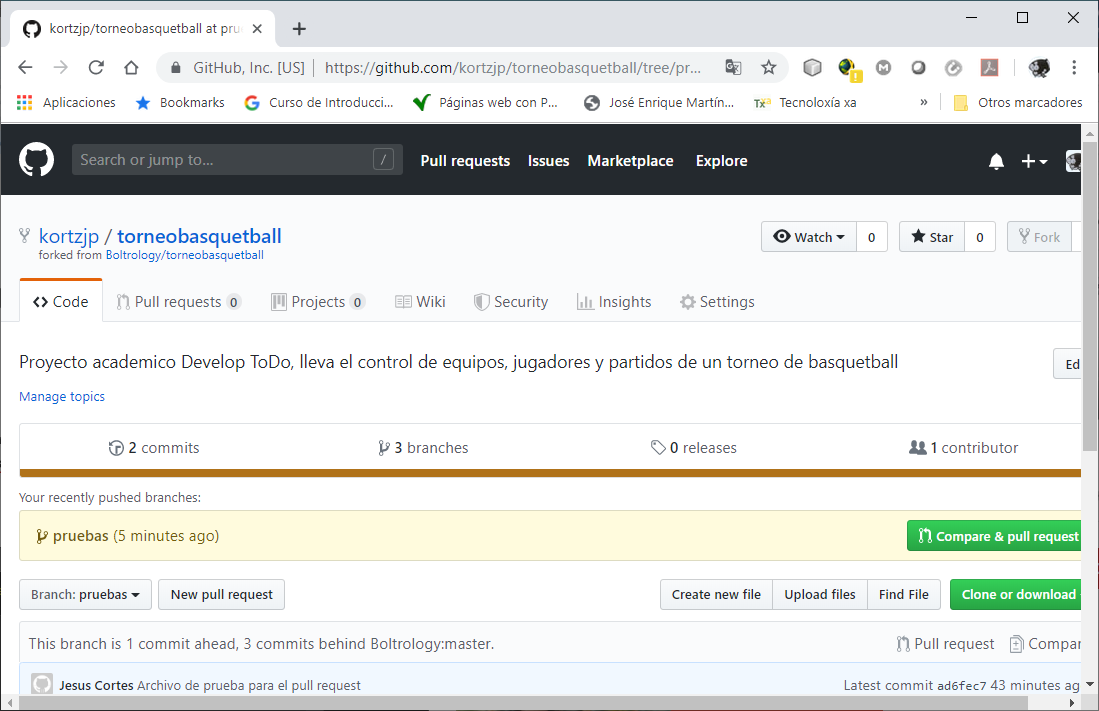
1. Con esto ahora podemos solicitar a desarrollador dueño del repositorio original acepte nuestra contribución, para esto primero tenemos que enviar nuestra rama local(pruebas) al fork para eso usamos el siguiente comando **git push [remoto\_recibe] [remoto\_envia]** para nuestro caso remoto\_recibe es **fork**  y remoto\_envia es **pruebas**

****

1. como se observa en el resultado del push se ha creado una nueva rama, la cual podemos observar en el navegador web

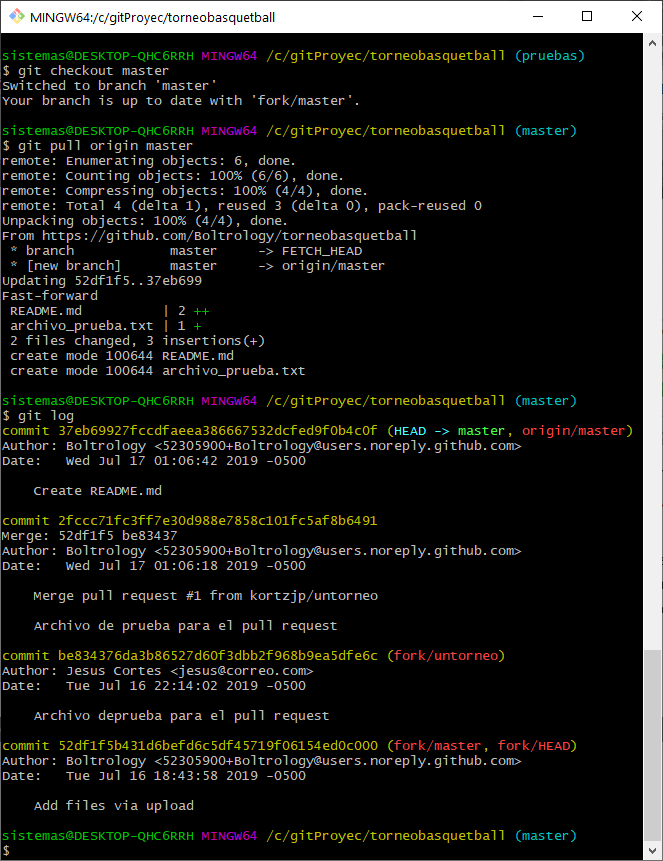


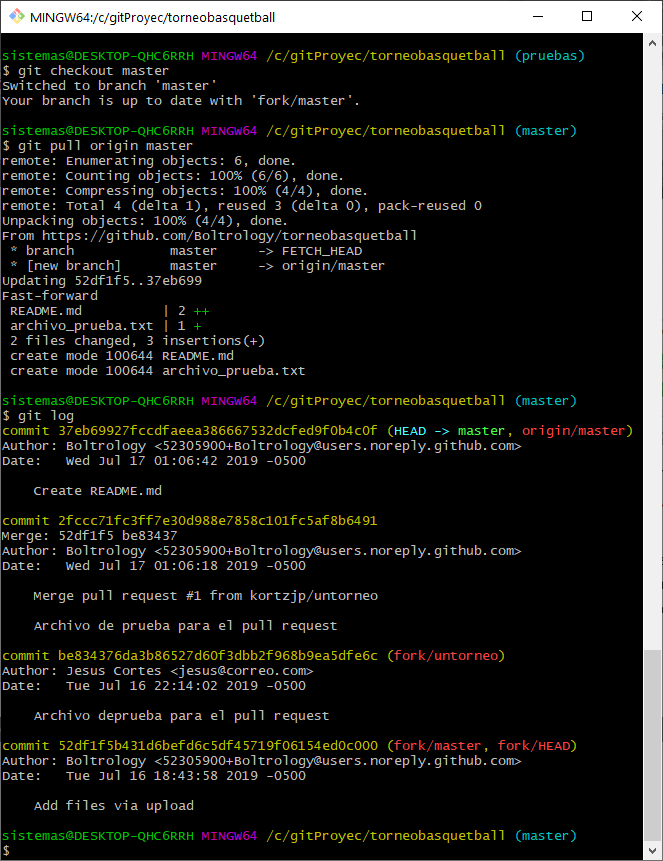
1. Al acceder a la nueva rama ahora podemos solicitar al dueño del repositorio que acepte nuestra contribución dando clic en el botón **new pull request**



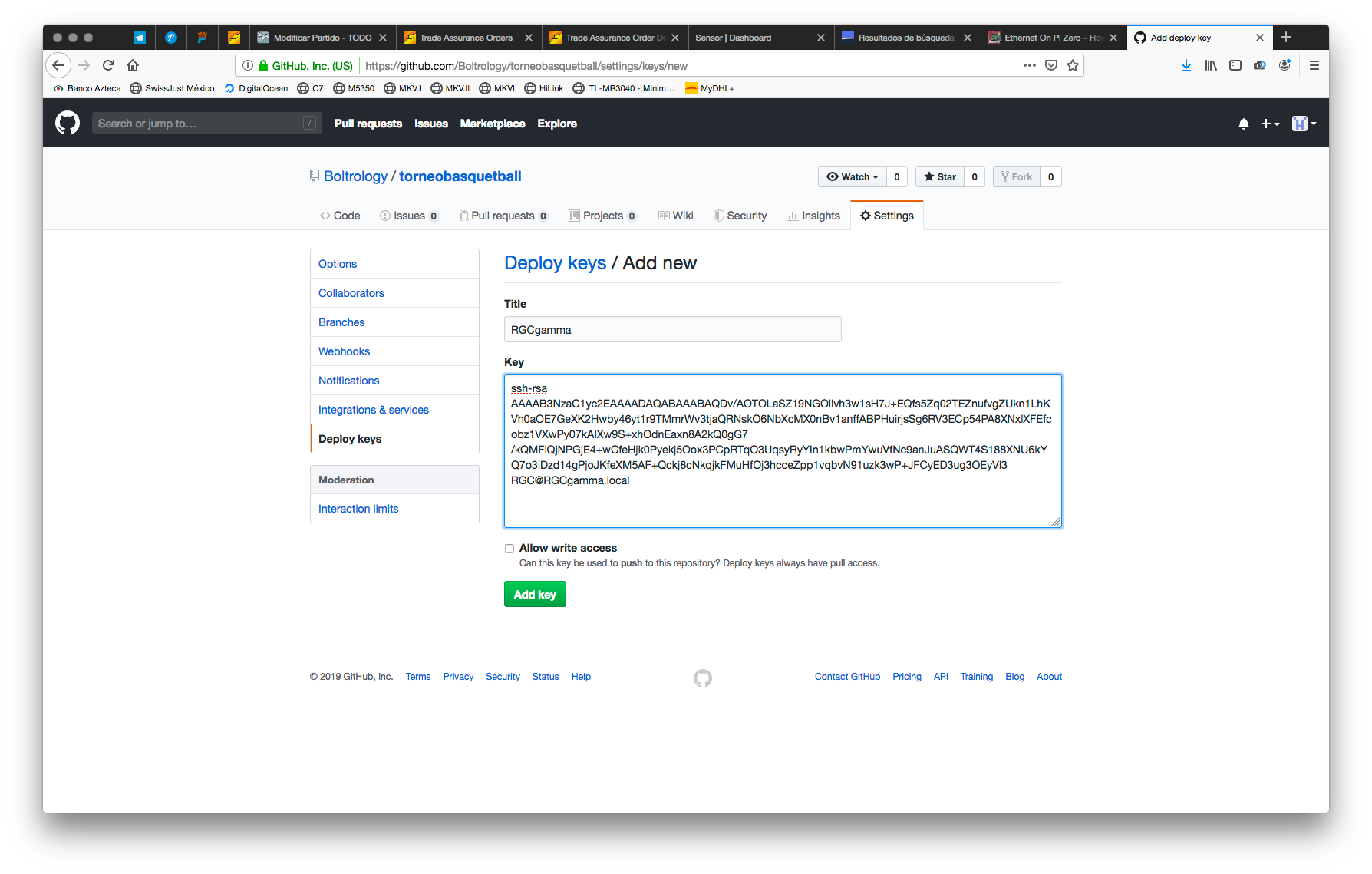
1. Después nos aparece un formulario con los datos del push y la opción para agregar un comentario a nuestra solicitud de pull request, por ultimo damos clic en **Create pull request**  y solo queda esperar a que el dueño acepte o no nuestra contribución.

Por último si el dueño del repositorio acepto las modificaciones, nuestro repositorio master ahora estará desactualizado, para solucionar esto dentro de la rama master realizamos un **git pull origin** master para que se actualice y después con un **git log** podemos ver nuestras contribuciones en los commit



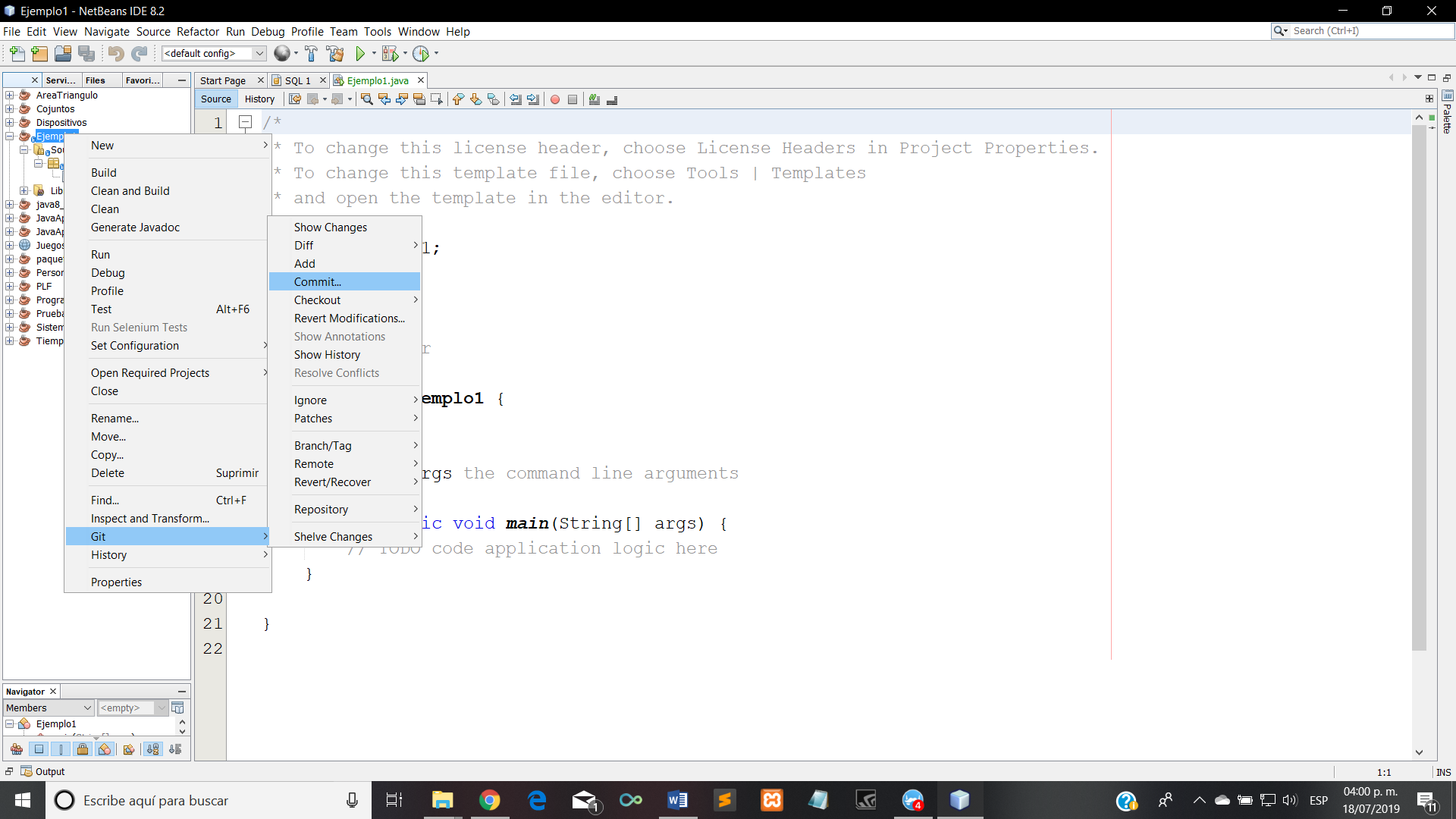


Creación de un proyecto usando la interface web de Github.

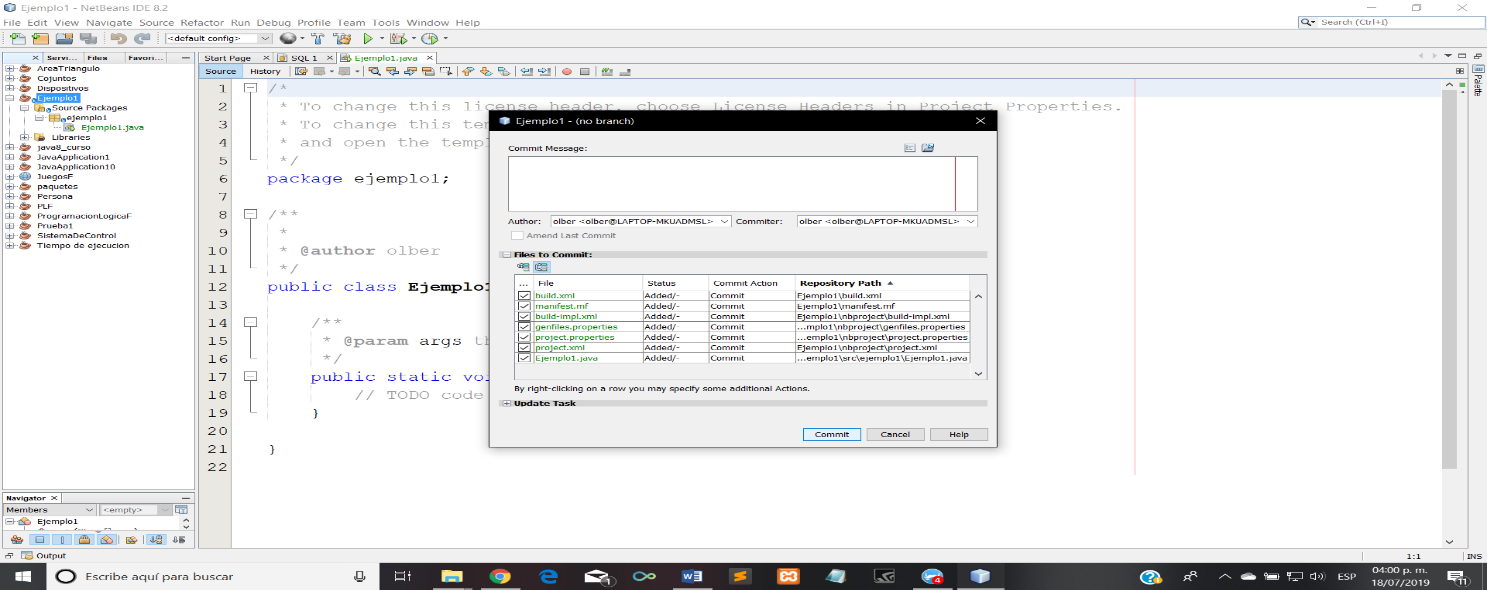
1. Creando un proyecto.
   1. Nombre del proyecto, se debe poner el nombre del proyecto, este debe ser único
   2. Descripción del proyecto
   3. Tipo del proyecto, este puede ser:
      1. **Publico**, que todos lo puedan ver
      2. **Privado**, solo un grupo de usuario lo puede ver.
   4. Si tienen archivos que no quieres en el git, por ejemplo archivos de compilación, o documentación personal.
2. Agregando archivos, primer commit (Web)
   1. Selecciona los archivos
   2. Arrastra los archivos, solo para usuarios que sean participantes.
   3. La pantalla debe ser esta:
3. Agregando archivos (SSH)
   1. Crea tu llave, de preferencia con el formato: id\_rsa.[nombre]
   2. Ingresa donde se va a guardar
   3. Clave de la llave
   4. Copia el archivo: id\_rsa.[nombre].pub
   5. Ve a Settings > SSH and GPG Keys > Add new
   6. Nombra la llave y pega en donde dice Key:
   7. Luego desde linea de comando configura en .ssh/config:
      1. Ubicación de la llave
      2. Nombre de usuario
   8. Clonando proyecto y haciendo un cambio
      1. Para obtener el proyecto se usa el siguiente comando: git clone git@github.com:[usuario]/[proyecto].git
      2. Para entrar al proyecto y poder usarlo: cd [proyecto]
      3. Se puede realizar los cambio necesarios, se coloca un ejemplo simple.
      4. Para hacer un cambio se usa el comando: git commit -m [mensaje]. Este debe ser especifico indicando los cambios y las clases que fueron modificadas.
      5. Para que el cambio se aplicado en GitHub debe ejecutarse el comando: git push git@github.com:[usuario]/[proyecto].git ó simplemente “git push”
4. “Pull request”, aplicando un cambio de otro participante.
   1. Seleccionar la pestaña “Pull request”
   2. Elegir la solicitud de actualización
   3. Una vez elegida la actualización, se puede confirmar y complementar el comentario.
   4. O cancelarla y también agregar un comentario

Subir un cambio via web y NetBeans.

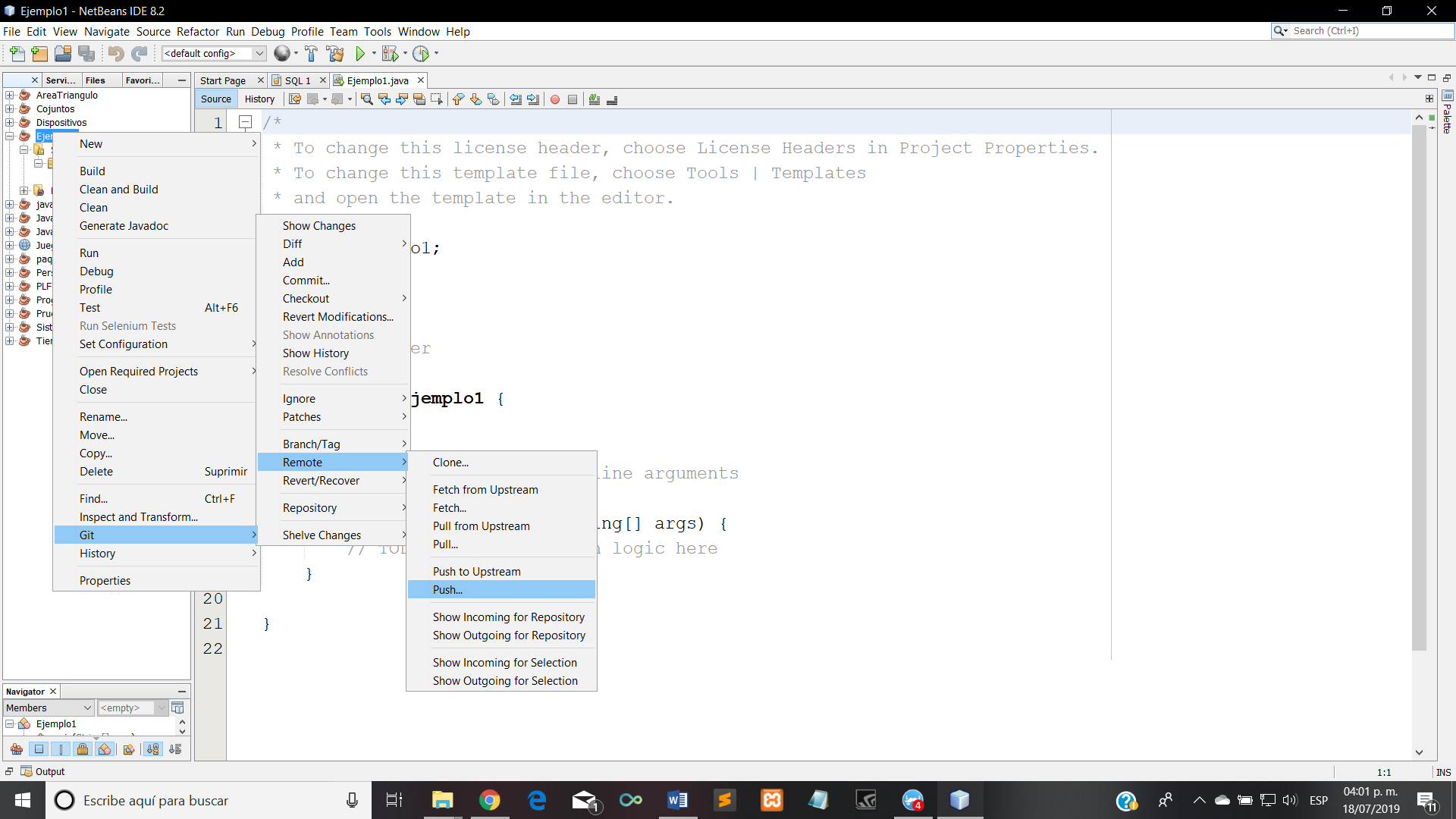
1. Para subir un archivo de NetBeans a via web
   1. Click derecho sobre nuestro proyecto opción Git seguido Commit …



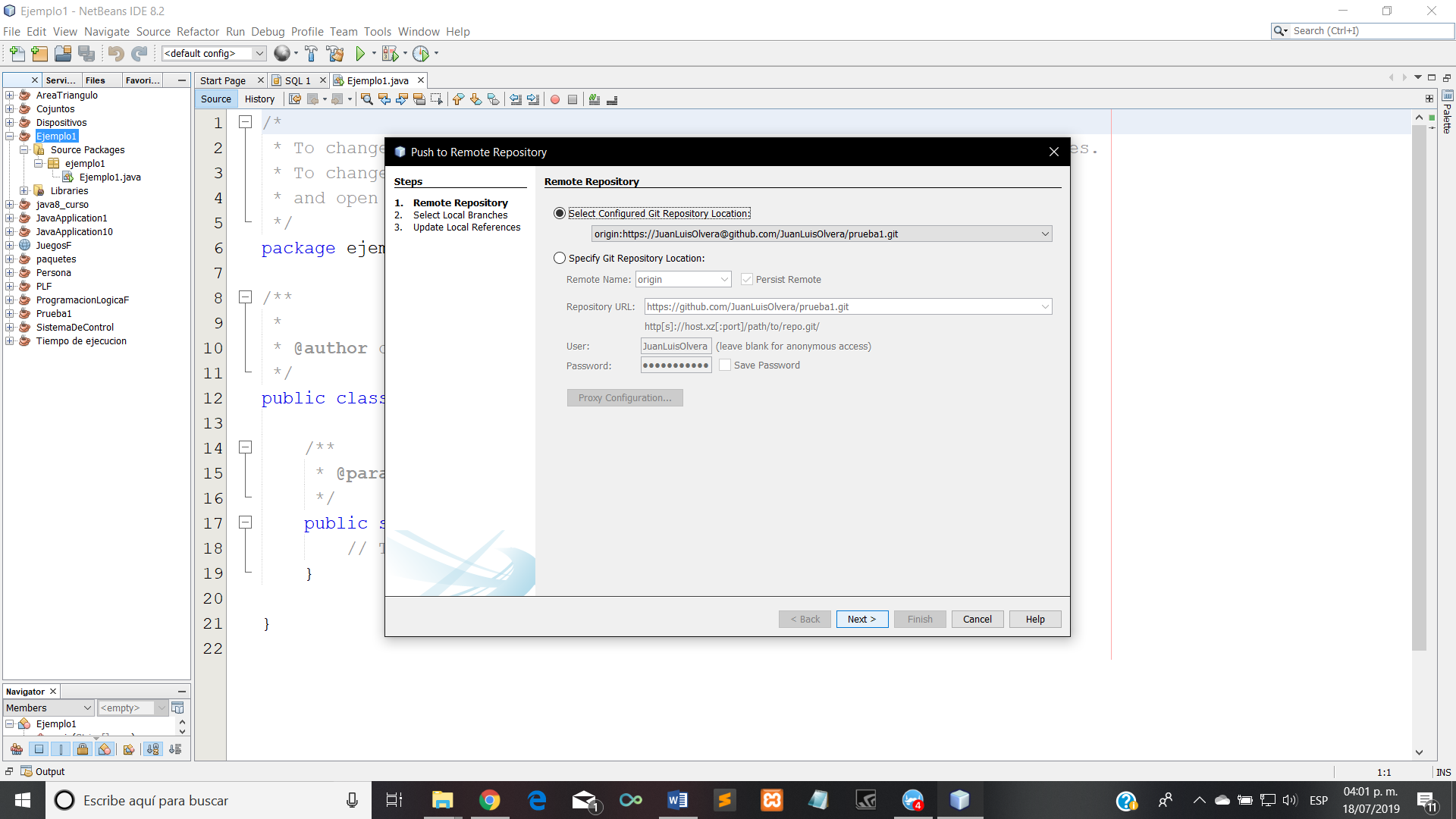
1. En la ventana emergente seleccionamos opción Commit



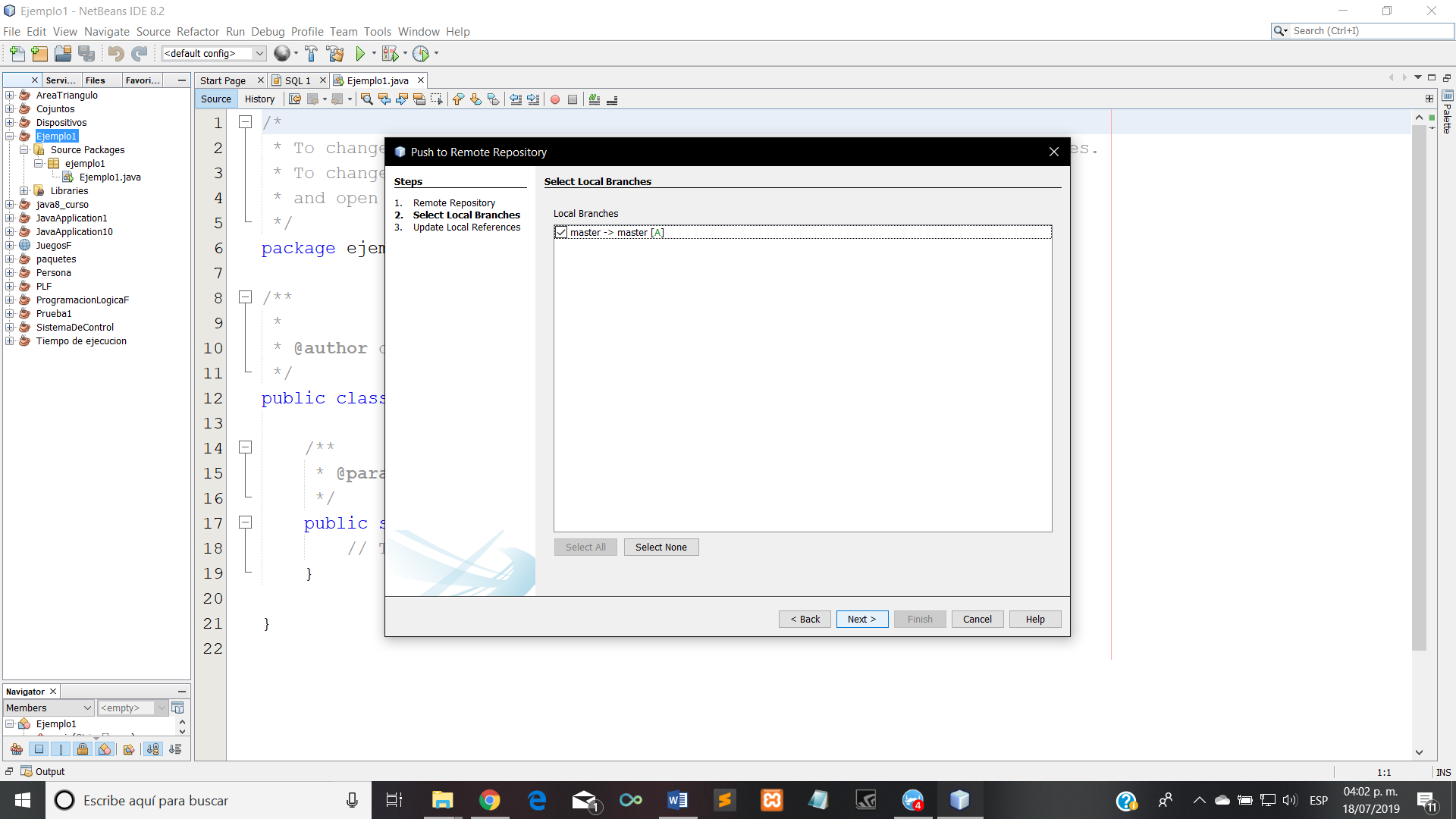
1. Para actualizar los cambios hechos en NetBeans a Git
   1. Click derecho sobre nuestro proyecto Git, Remote y push



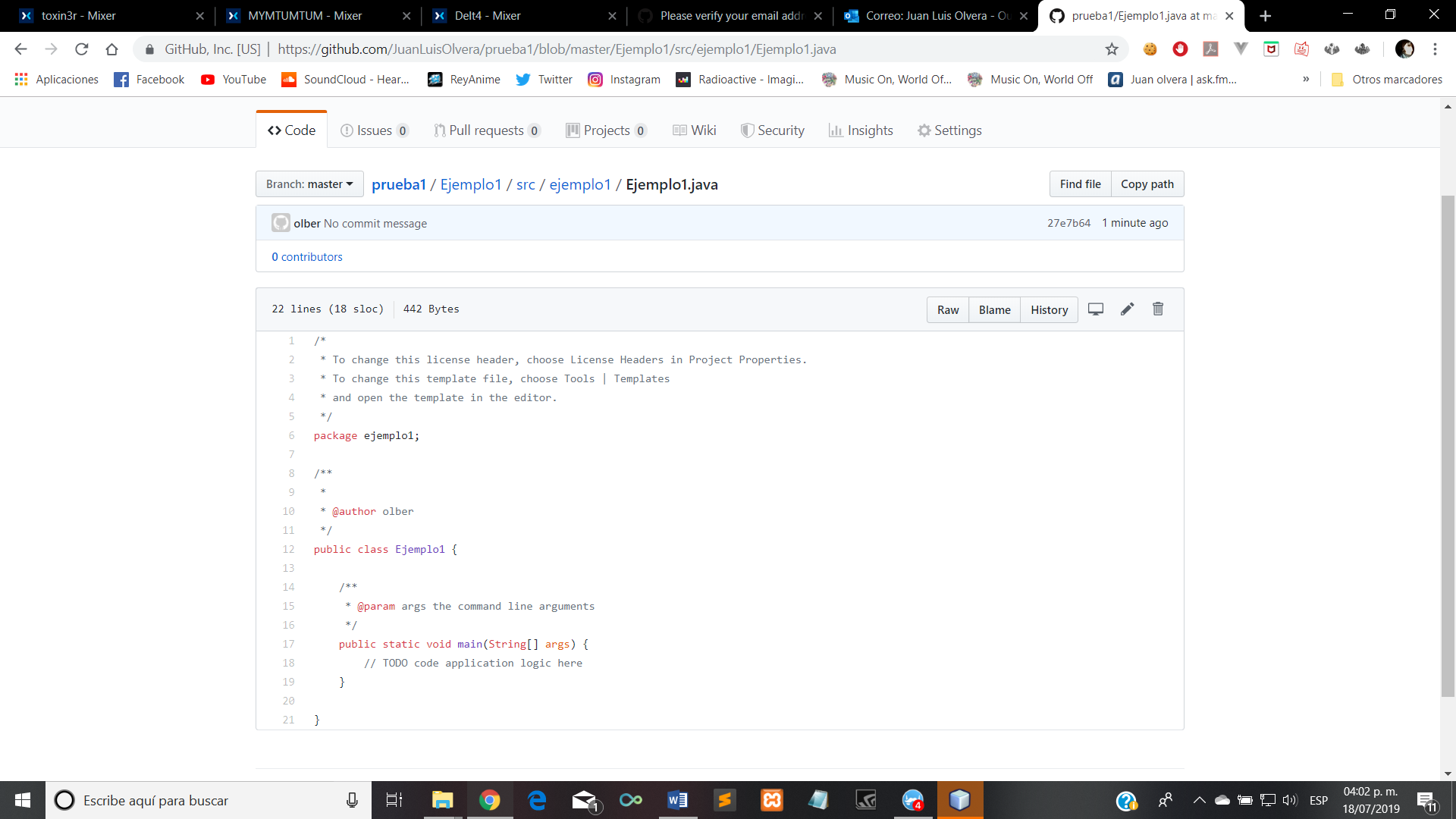
1. seleccionamos la opción de next de la ventana emergente



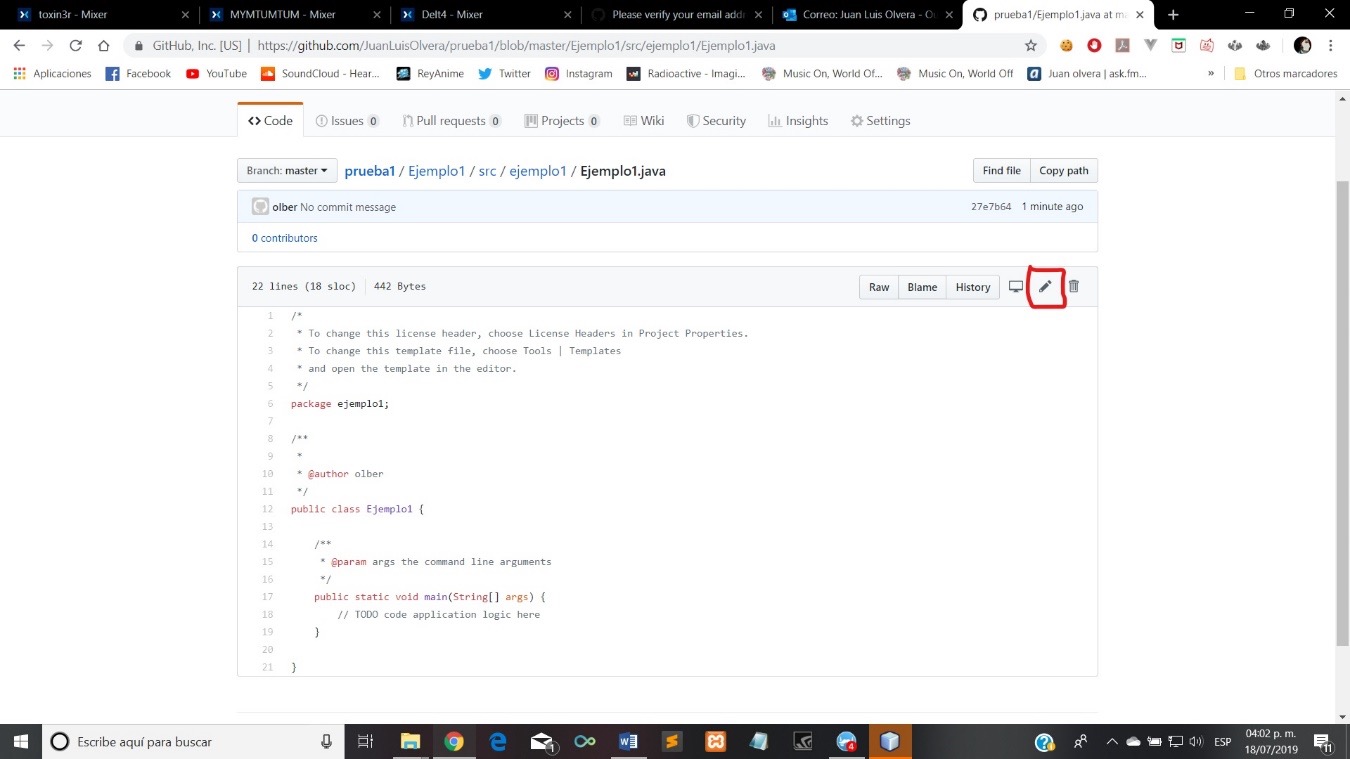
1. Palomeamos la opción de Master y next

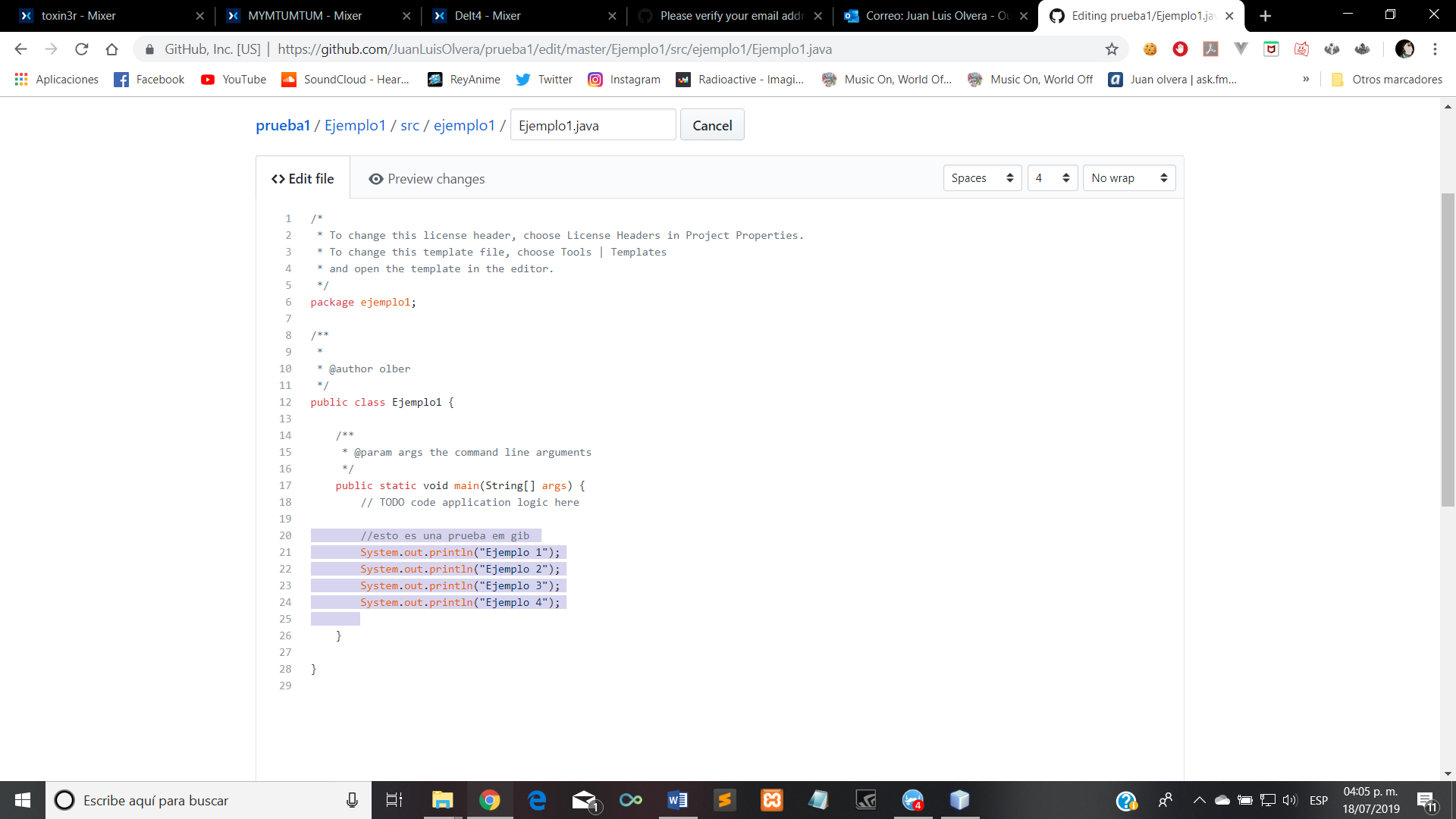


1. Vamos a nuestra pág. de Web y refrescamos

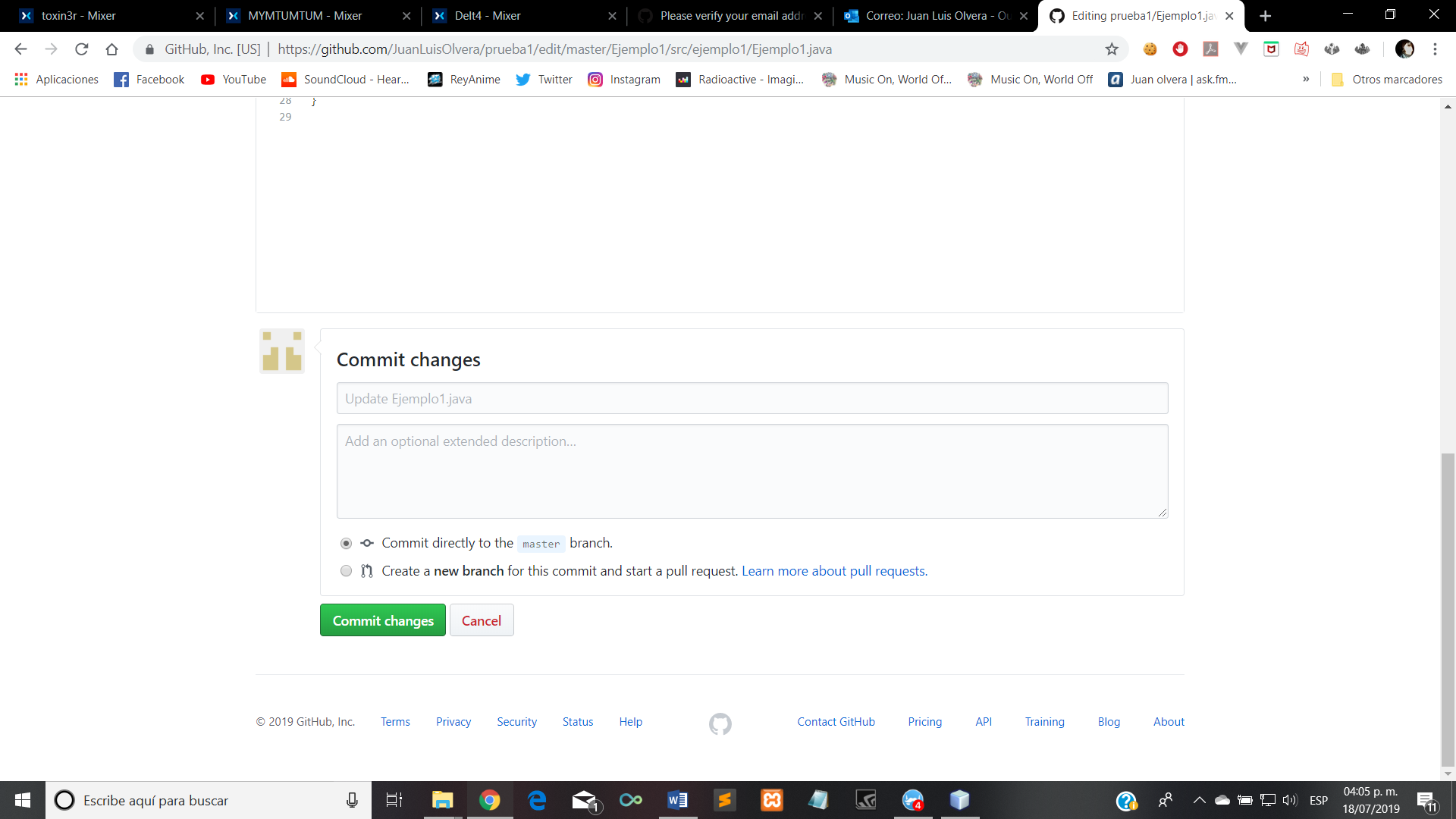


1. Para editar el código desde via web
   1. Seleccionamos la opción del lápiz y procedemos a editar el código

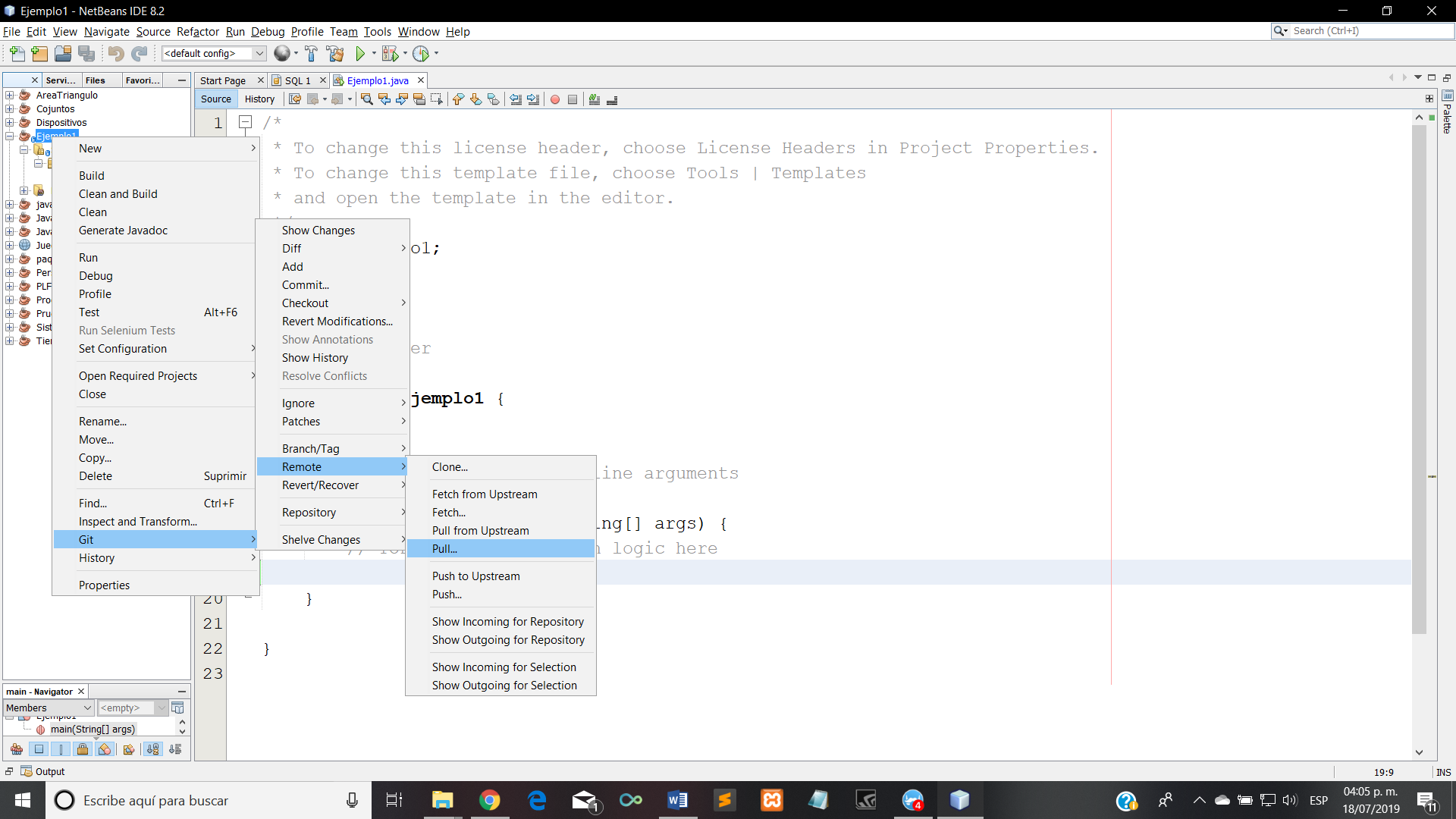




1. Para guardar los cambios vamos hasta la parte inferior del editor y seleccionamos Commit Changes



1. Para actualizar los cambios hechos de el editor web y se reflejen en NetBeans
   1. Vamos a NetBeans y damos clik derecho a nuestro proyecto Git, Remote, pull



1. Seleccionamos next en la ventana emergente y los cambios del editor web se descargarán en nuestro netBeans
2. Cambios reflejados

