Versionado

Una vez que iniciamos un proyecto en algún momento necesitaremos hacer cambios en el y tal vez (seguramente) en algún momento nos veamos obligados a regresar a un punto anterior, una posible solución sea copiar nuestro proyecto entero con un nombre distinto, si somos un poco precavidos quizás el nuevo nombre sea la fecha y hora en que creamos esa copia, pero terminar con un montón de carpetas con nombres similares es poco práctico, después podríamos recurrir a alguna herramienta de versionado como SVN que nos ayudara a tener un registro de los cambios de nuestro proyecto, pero herramientas como esta depende en gran parte de una conexión para guardar nuestros cambios y así es como llegamos a Git, una herramienta que nos permite guardar una copia de nuestros archivos, pero todo la teoría la podrás leer en el [libro de git](https://git-scm.com/book/es/v2)

Ahora presta atención. Esto es lo más importante que debes recordar acerca de Git si quieres que el resto de tu proceso de aprendizaje prosiga sin problemas. Git tiene tres estados principales en los que se pueden encontrar tus archivos:

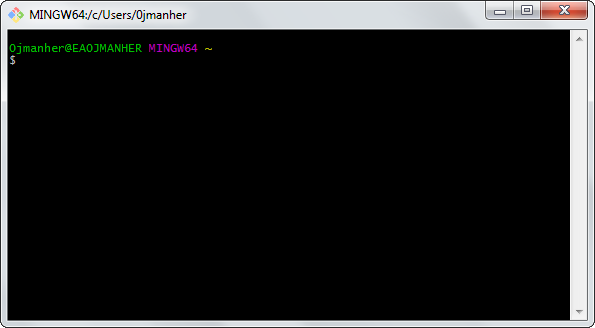
* modificado (modified): significa que has modificado el archivo pero todavía no lo has confirmado a tu base de datos.
* preparado (staged): significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.
* confirmado (committed): significa que los datos están almacenados de manera segura en tu base de datos local.

El flujo de trabajo básico en Git es algo así:

* Modificas archivos en tu directorio de trabajo.
* Preparas los archivos, añadiéndolos a tu área de preparación.
* Confirmas los cambios, lo que toma los archivos tal y como están en el área de preparación y almacena esa copia instantánea de manera permanente en tu directorio de Git.

Ya iniciados en esto tengo que avisarte que trabajaremos con git por medio de la consola, ya que como lo mencionan en el libro si sabes usar los comandos sabrás usar las herramientas graficas pero inversamente podría no ser igual.

La instalación en Windows es realmente sencilla, puedes descargarlos [aquí](https://git-scm.com/), las imágenes si las requieres las dejare en una carpeta a un lado. Bien una vez instalado. En el inicio buscamos “git bash”:



Se abrirá una ventana como esta y sabremos que la instalación fue correcta, ahora necesitamos un lugar donde guardar nuestro código para ellos debemos tener una cuenta en [github](https://github.com/), que es el que usaremos por el momento, antes de poder hacer cualquier cosa regresemos a nuestra consola de git bash y ejecutamos lo siguiente:

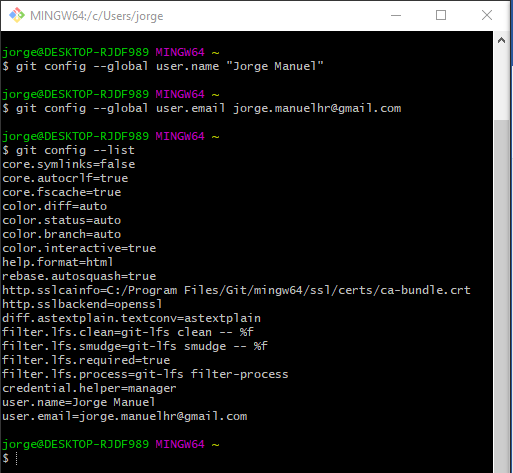
git config --global user.name "TU\_NOMBRE\_ENTRE\_COMILLAS"

Damos enter y enseguida ejecutamos:

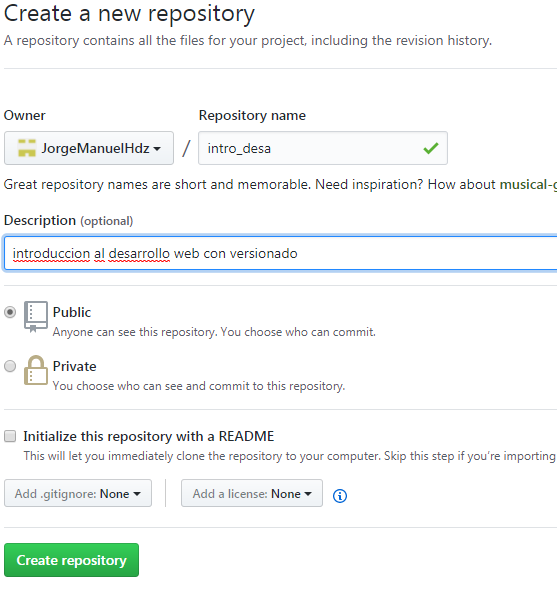
git config --global user.email TU\_CORREO\_SIN COMILLAS

Bien si llegamos a equivocarnos o queremos cambiar algún dato solo hay que correr el comando con el valor que queramos, para ver cuáles son los valores guardados solo hay que ejecutar el comando:

git config --list



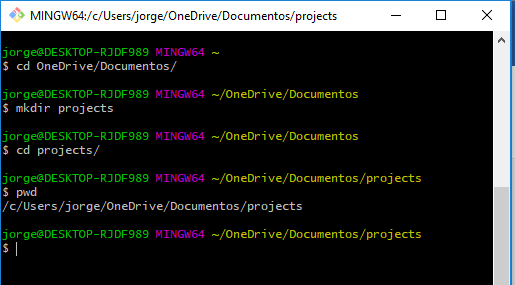
En github buscamos la opción de “new repository”, y llenamos los datos requeridos:



Hemos creado un nuevo repositorio, busca la siguiente imagen y fija en SSH, copia al portapapeles:

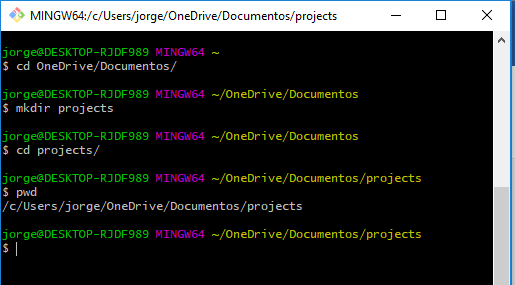


Al momento de hacer el tutorial me encuentro en Windows 10 por lo que creare mi carpeta dentro de OneDrive/Documentos, tú puedes crearla donde gustes, en Windows 7 y otras puede ser directamente en la carpeta Documentos, después ubicaremos nuestra consola en esa carpeta



Expliquemos los comandos básicos que usamos:

* pwd: print work directory, nos mostrara la ubicación actual en nuestra terminal(consola)
* cd: change directory, es para cambiar nuestro directorio, este comando requiere otro argumento el cual es la dirección a la que se moverá desde la ubicación actual, estos argumentos son
  + ~ el signo sobre la eñe nos lleva al home o la ubicación con la que inicia la terminal
  + .. los dos puntos nos regresan una carpeta desde nuestra ubicación actual
  + “cd D:” la letra D indica a cual disco queremos cambiarnos, esto en caso de que queramos guardar nuestros proyectos en un disco externo o alguna partición de nuestro disco duro.
  + Para entrar a una carpeta bastara con poner el nombre de la carpeta con mayúsculas y minúsculas en caso de que la carpeta tenga espacios el nombre debe de la carpeta debe estar entre comillas
* ls: muestra los archivos en nuestra ubicación, directorios(carpetas) y archivos.
* clear: limpia nuestra consola.

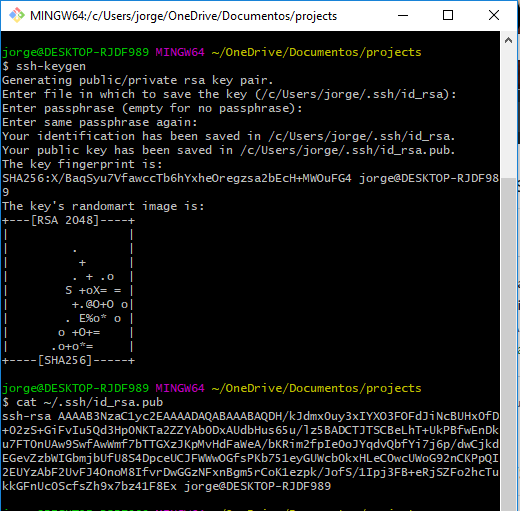


Puedes ver como después de que ejecute “cd OneDrive/Documentos” cambia la parte después de MINGW64 mostrándonos la ubicación en la que nos encontramos.

Ahora correremos los siguientes comandos:

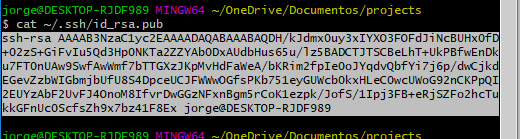
1. ssh-keygen
   * al dar enter nos saldrá algo como “Enter file in which to save the key” aquí simplemente damos enter
   * después nos pregunta una frase “Enter passphrase (empty for no passphrase):”, escribiremos una frase que cualquiera,veremos que no se muestra lo que tecleamos y al finalizar presionamos enter y volvemos a teclear la misma frase.
2. cat ~/.ssh/id\_rsa.pub

se verá algo como esto:

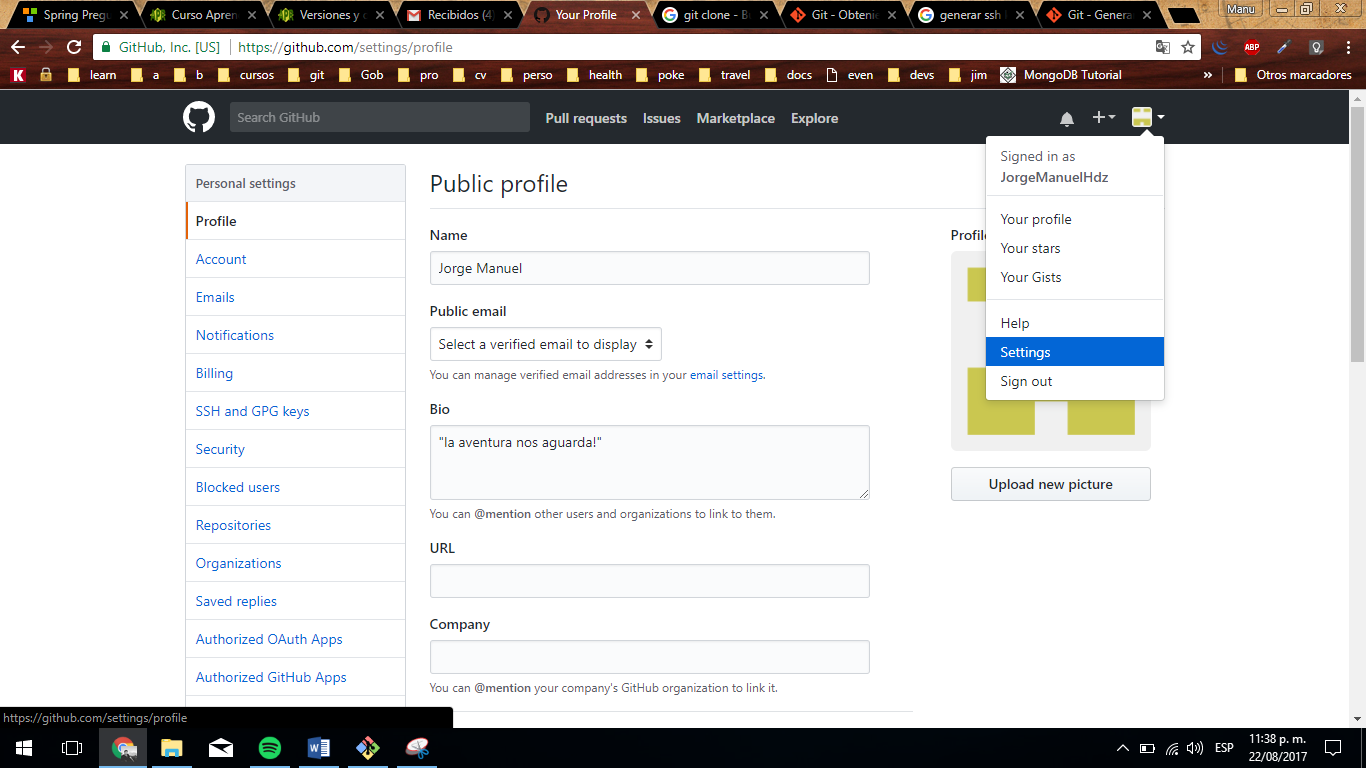


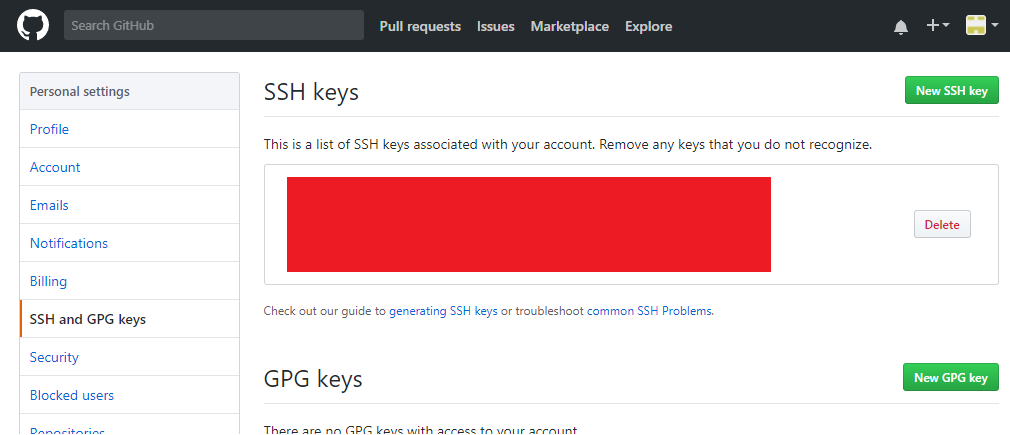
Felicidades, acabas de crear una llave ssh publica, por ahora nos preocuparemos mucho, con el segundo comando siempre veremos nuestro archivo pub que lo comprenderas conforme te familiarices con la consola, la ubicación de nuestras llaves se ubicaran en la primer pregunta que nos hace después del primer comando en mi caso mis llaves se encuentran en “/c/Users/jorge/.ssh/id\_rsa”.

De nuestro segundo comando copiamos lo que nos muestra “cat”

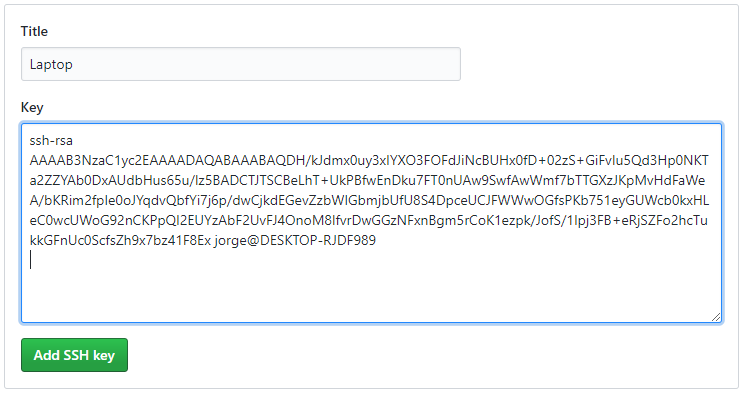


En github vamos a settings > SSH and GPG keys > New SSH Key

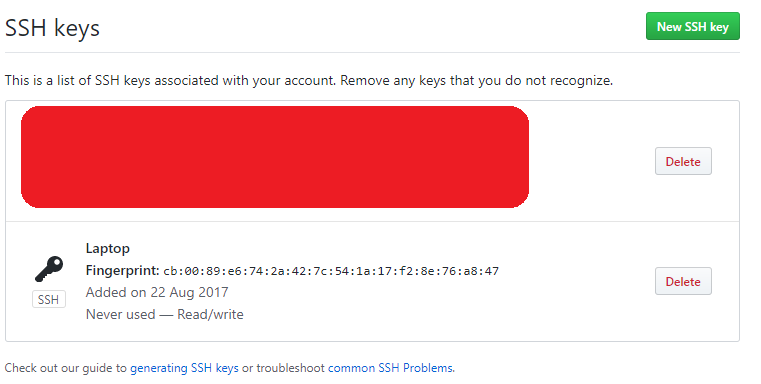




En title escribimos la descripción del equipo que estamos usando, y debajo pegamos lo que habíamos seleccionado y agregamos la llave( Add SSH Key)



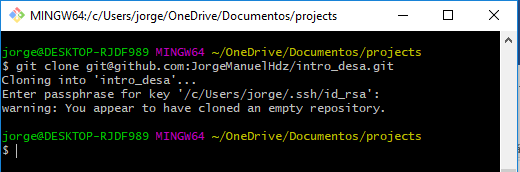
Asi es como podremos agregar todos los equipos desde los que queremos trabajar de forma segura:



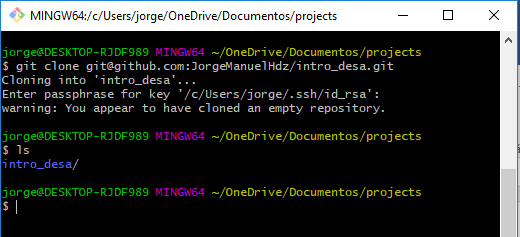
Recuerdas la imagen anterior:



Bien vuelve a “copy to clipboard”(copiar al portapapeles), regresando a nuestra consola ejecutamos el comando “git clone” y dejando un espacio pegaremos lo que acabamos de copiar.

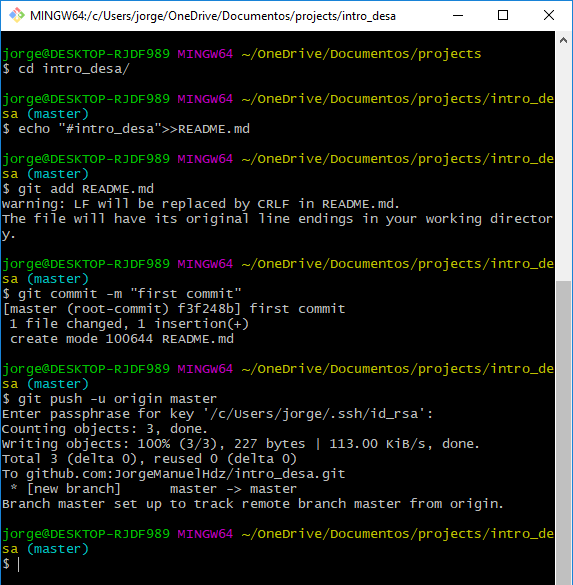


Veras que nos pide la frase de cuando creamos nuestra llave SSH, la escribiremos y nos mostrara una pequeña advertencia ya que nuestro repositorio está vacío. Al listar(ls) veremos que nos muestra el repositorio que creamos en github



Bien pues hagamos nuestro primer push. ¡Y todo desde nuestra consola!!

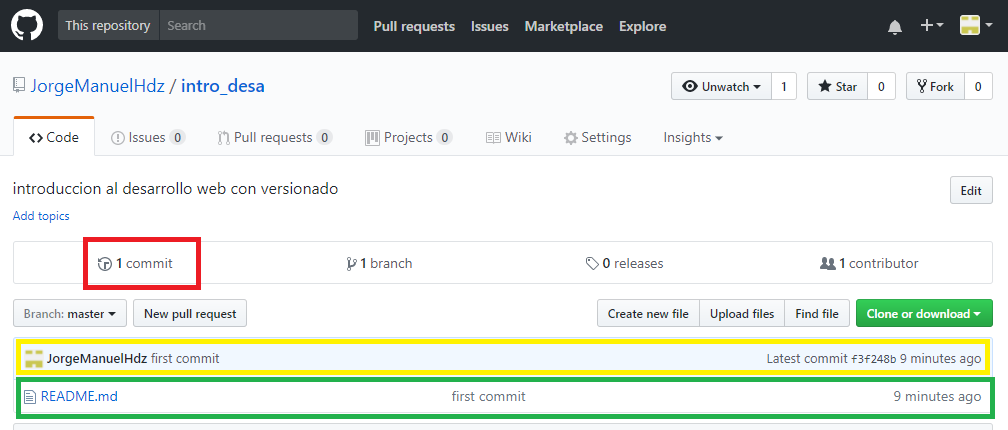
1. cd intro\_desa/
2. echo "# intro\_desa" >> README.md
3. git add README.md
4. git commit -m "first commit "
5. git push -u origin master



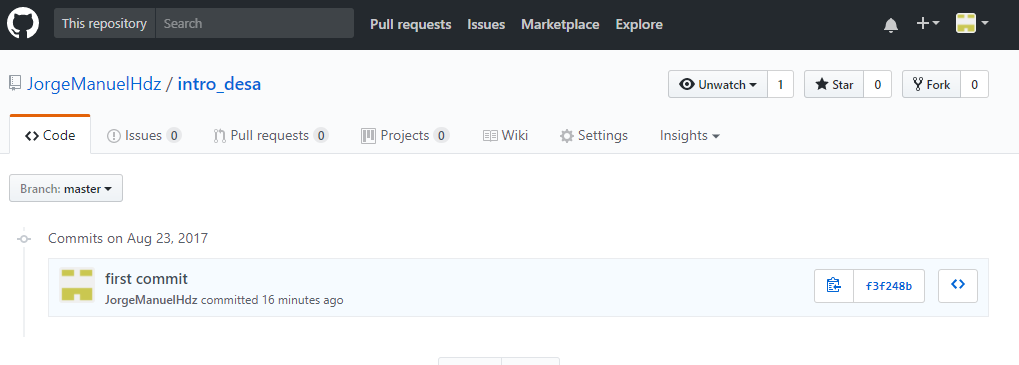
La explicación de lo que hicimos:

1. entramos en nuestra carpeta “intro\_desa”
2. creamos un archivo llamado “README.md” y dentro de ese archivo hemos escrito “#intro\_desa”
3. indicamos que queremos versionar “README.md”
4. agregamos un mensaje para los cambios que guardaremos, en este caso agregaremos un archivo solo para indicar que es nuestro primer commit
5. enviamos todo a nuestro repositorio (BD con los cambios de código), aquí deberemos escribir la frase que pusimos al crear nuestra llave SSH

bien si vamos a nuestra cuenta de GitHub podremos ver los cambios que hemos hecho:



En la parte amarilla veremos el ultimo commit y quien lo realizo, también veremos cuando se realizó, en la parte verde veremos los archivos junto con su ultimo commit y cuando se realizó el cambio. En la parte roja veremos todos los commits que hemos realizado, si damos click podremos verlos con más detalle.



Si damos click sobre el mensaje del commit en este caso “first commit” podremos ver los cambios que se realizaron en el commit.



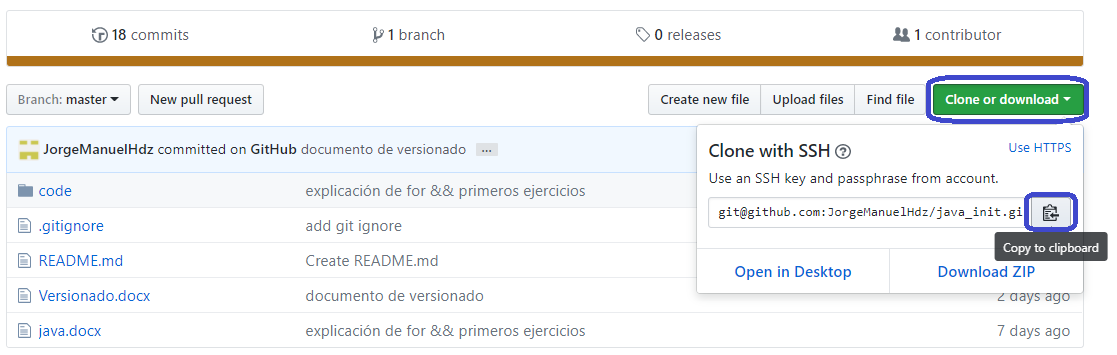
Ahí nos indicara que cambio un archivo, que se agrego una línea y se eliminaron 0 lineas.

Podremos ver que al lado de “#intro\_desa” aparece un “+” que indica que es la línea que fue agregada.

Clonando un repositorio

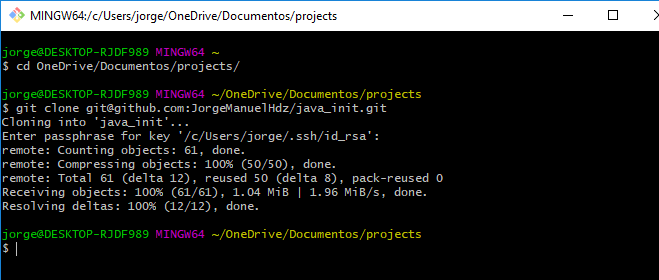
Ya hemos visto como crear un repositorio, ahora veremos cómo trabajar sobre uno ya existente, a esto se le llama clonar un repositorio.

Primero ingresamos a un proyecto y buscamos la opción “clone or download” y copiamos la liga del repositorio.

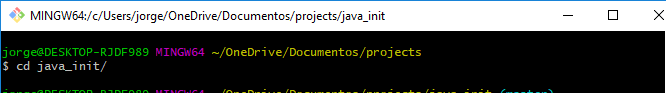


Después en nuestra consola entramos a nuestra carpeta de proyectos esto lo hacemos con

1. cd [ruta de carpeta] en mi caso OneDrive/Documentos/projects
2. git clone [pegamos lo que copiamos anteriormente]



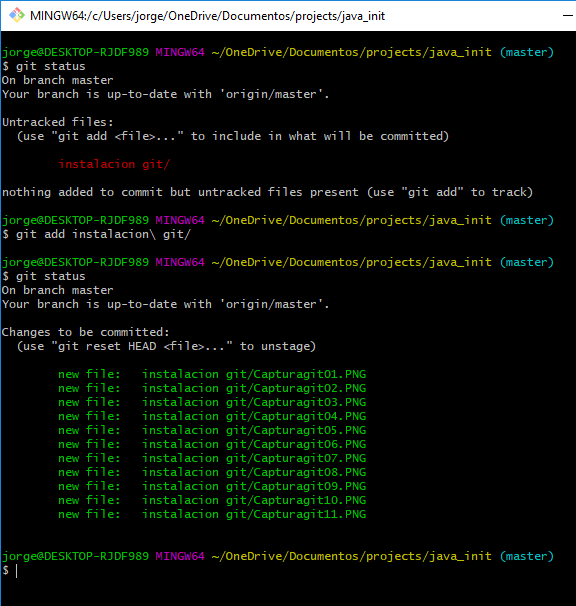
Ingresamos a la carpeta, en este caso el nombre de la carpeta se encuentra la final de la liga que habíamos copiado, que es el nombre del proyectoes decir “java\_init”

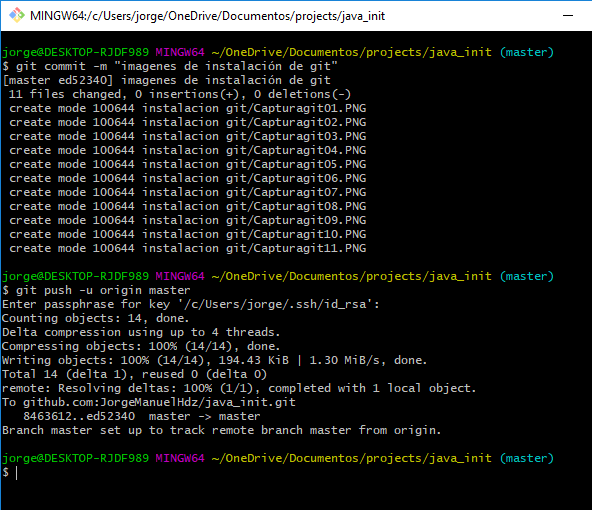


En esta parte podemos hacer todos los cambios en el código que queramos, cuando decidamos que es momento de guardar nuestro avance antes de intentar algo nuevo haremos algo asi, para este punto recuerda que git maneja 3 estados, modificado, preparado y confirmado, pero incluso antes de cualquiera de estos hay que indicar que queremos guardar los cambios que le vayamos haciendo a un documento, a eso se le llama versionar.

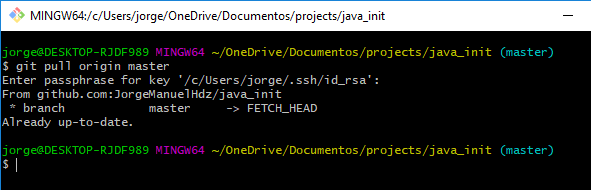
Los comando que usaremos son:

1. git status: nos indicara el estus de nuestros archivos, el rojo indica que no esta versionado
2. git add [archivo]: indicaremos cual archivo queremos versionar, en este caso es una carpeta con imágenes
3. git status: para ver como quedaron nuestros cambios, al estar verde quiere decir que nuestros archivos están “modificados”.
4. Git commit -m “[mensaje descriptivo de cambios realizados]”: preparamos nuestros cambios.
5. Git push -u origin [DONDE]: confirmamos que deseamos guardar nuestros cambios, por ahora la ubicación que puedes usar es “master”



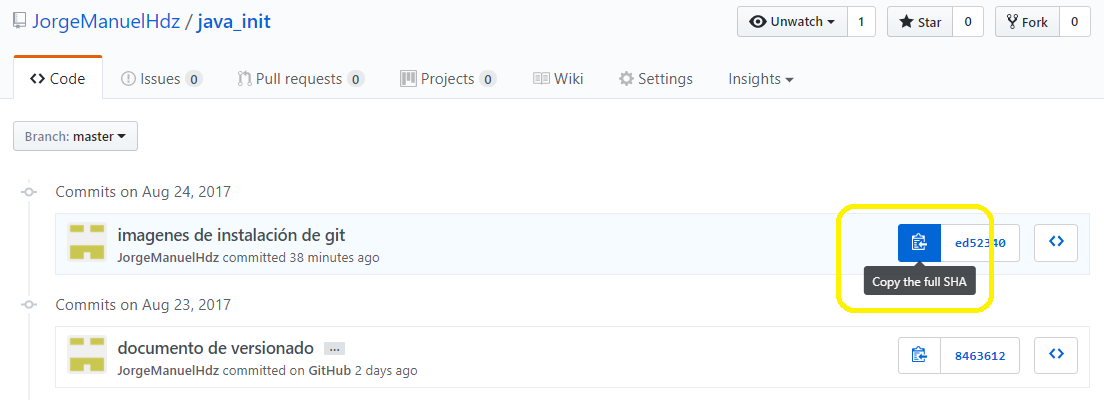


Algo importante que deberás hacer muy seguido es ejecutar el comando “git pull origin [DONDE]” para descargar los cambios que otros usuarios vayan subiendo a ese proyecto.

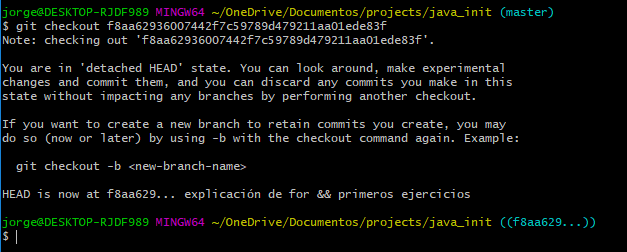


Otro comando que te será muy útil será “git checkout [COMMIT]”, recuerda que hay un apartado donde podemos ver los commits(cambios) que hemos realizado en nuestro repositorio, pues si trabajamos de forma ordena, podremos volver a ver como se encontraba el código en otros momentos, es decir, ¡VIAJAMOS EN EL TIEMPO!.

Solo hay que ir a la sección de commits buscar el cambio al que queramos regresar y copiar la llave SHA completa



Una vez que mas regresamos a nuestra consola(git bash) y ejecutamos “git checkout [commit seleccionado]



Podemos ver que al ejecutar el comando nos dice cuál es nuestro encabezado actual e incluso el mensaje del commit que tenía esa llave, y donde anteriormente nos decía “master” ahora nos muestra el principio de nuestra llave. Las llaves son únicas por lo que no pueden repetirse. Bien para poder hacer un cambio deberás regresar al encabezado por lo que debes correr nuevamente “git checkout master” y con esto podrás hacer commit nuevamente.

Resumen:

Git pull origin master: descargar cambios

Git push -u origin master: guardar cambios

Git status: ver estatus de archivos

Git add: preparar archivos modificados para ser guardados

Git commit -m “mensaje” indicar el mensaje para el commit