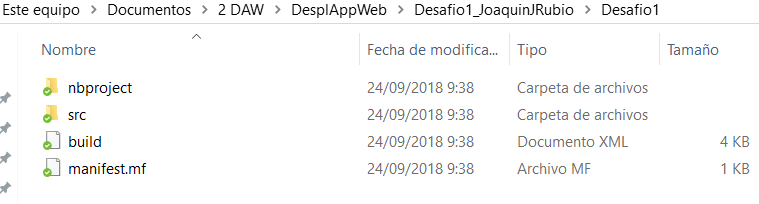
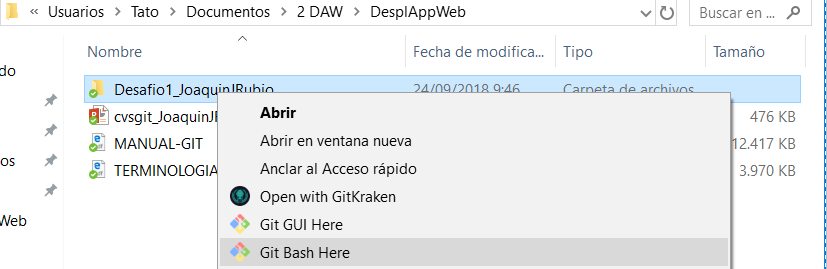
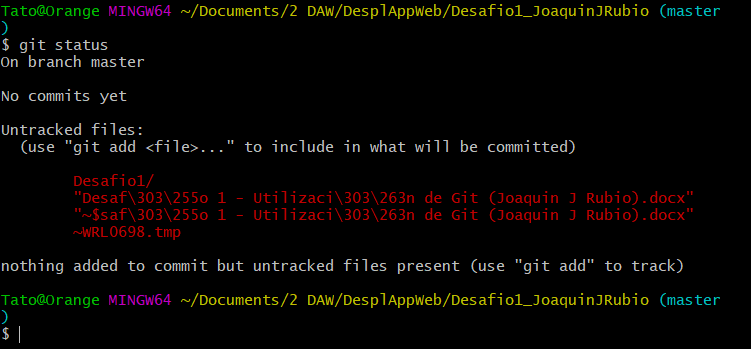
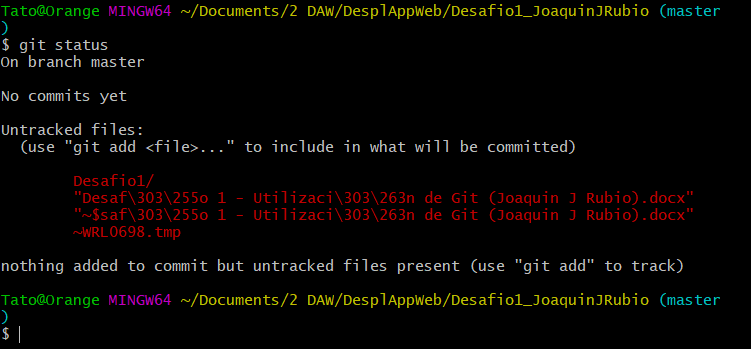
Desafío 1:

Después de la sencilla instalación de Git, y de un editor de código (NetBeans en mi caso), creamos un proyecto (en Java, por ejemplo), y lo ubicaremos en la carpeta de esta asignatura (DespAppWeb):



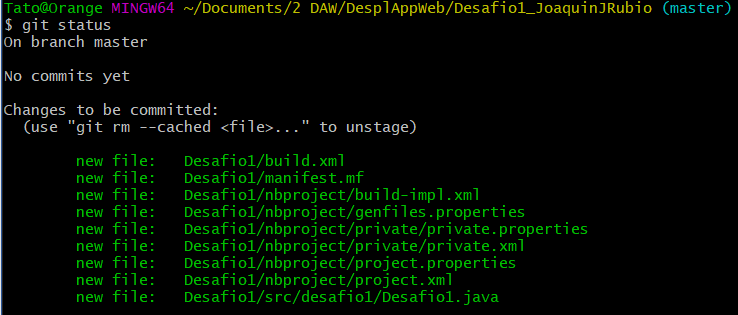
Creamos un nuevo repositorio desde el GUI de Git o desde Git en el terminal (en nuestro caso), para ello: Windows+R y escribimos “CMD”, nos dirigimos a la ruta anterior y ejecutamos el comando “**git init**”, una vez iniciado, dando botón derecho sobre la carpeta podremos ejecutar tanto la consola de Git como el GUI: 

Para comprobar los archivos que tenemos (en Git Bash Here): **git status**



Nos dice que hay archivos sin añadir y nos da consejo de como hacerlo.

Para añadir archivos: **git add Desafio/** y comprobamos que se ha realizado bien:

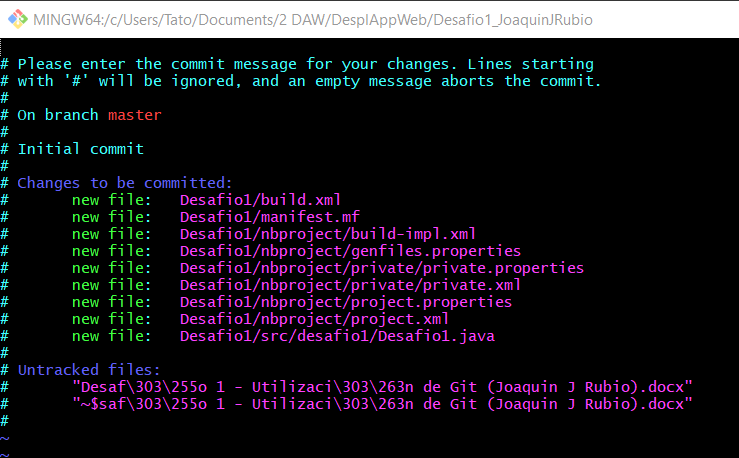


Antes de crear un commit, debemos introducir nuestra identidad, como medida de seguridad. Para ello usaremos:

* **git config –global user.email “joaquin@mail.com”**
* **git config –global user.name “joaquin”**

Primer snapshot del proyecto (commit): **git commit**

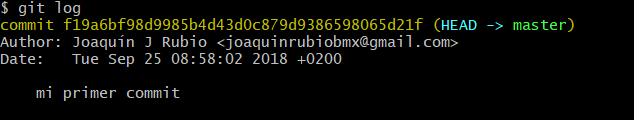
Si lo hemos realizado correctamente, se abrirá un documento como este:



Presionamos la letra ‘i’ para empezar a escribir, escribimos un mensaje para identificarlo, “escape, :qw”, aparecerá algo así, donde le da un hash/nº de identificación a cada archivo dentro del proyecto:

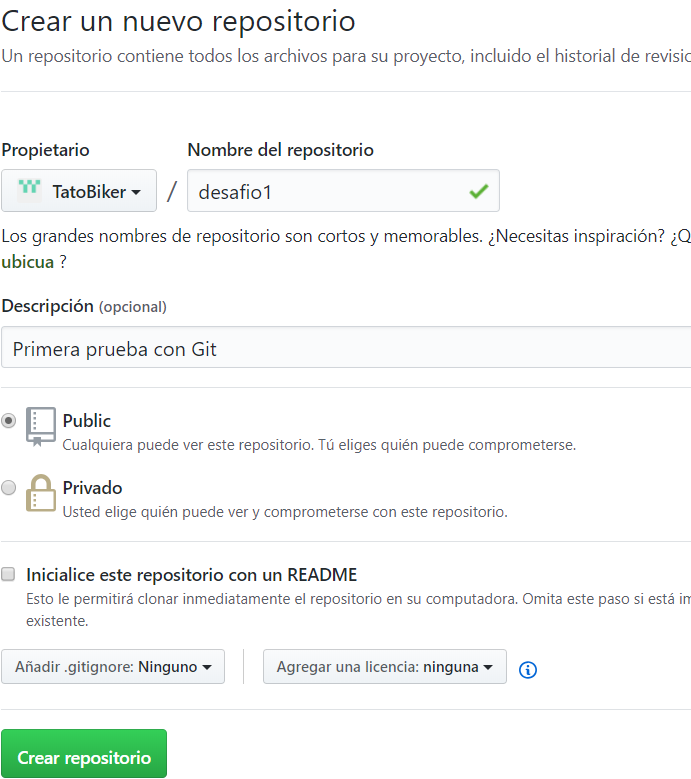


Con **git log**: para ver los commits realizados en el proyecto y mas información

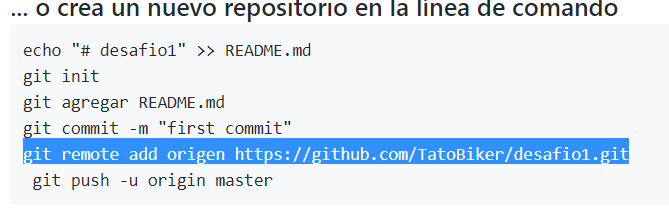


Si antes de hacer el siguiente commit, queremos no guardar los datos y recuperar los anteriores, podemos hacerlo con **git checkout – archivo.ext** y con **git dif** para ver las diferencias entre los archivos.

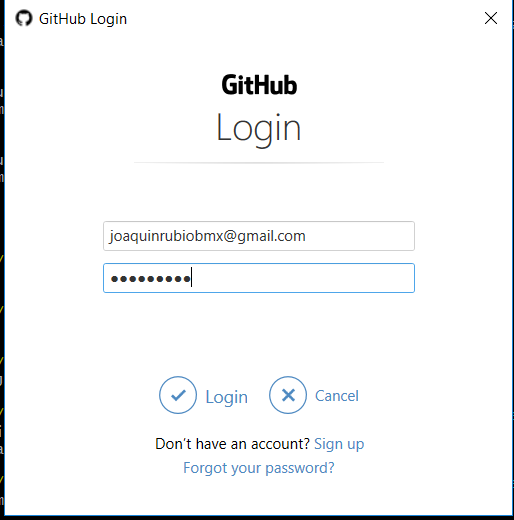
Entramos en [**https://github.com/**](https://github.com/)para acceder al repositorio, para ello creamos una cuenta y desde nuestro perfil > nuevo repositorio y rellenamos:



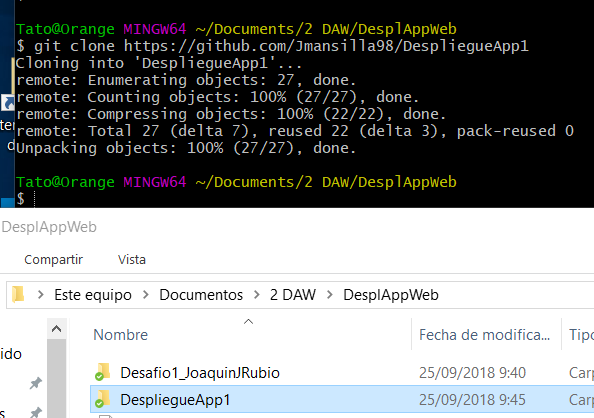
En la siguiente página al aceptar, copiamos la línea señalada y la ponemos en el editor:



Después escribimos la ultima línea, y si todo ha ido bien nos debe salir un login de GitHub, donde debemos introducir las credenciales de GitHub:



Por último, para clonar, utilizaremos el comando **git clone url\_compañero**, este es el resultado:

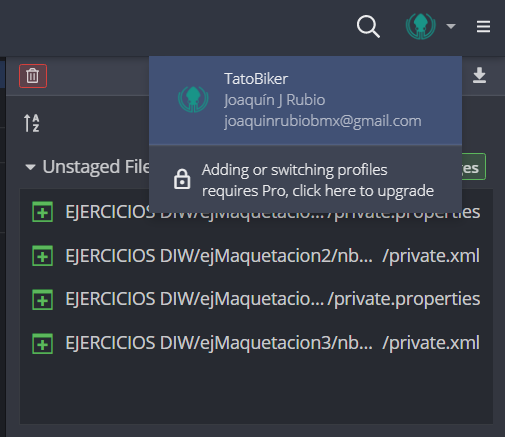


Para eliminar el repositorio (en mi caso no lo elimino para poder usarlo con GitKraken): “cd carpeta/” y ejecutamos **rm -rf .git**

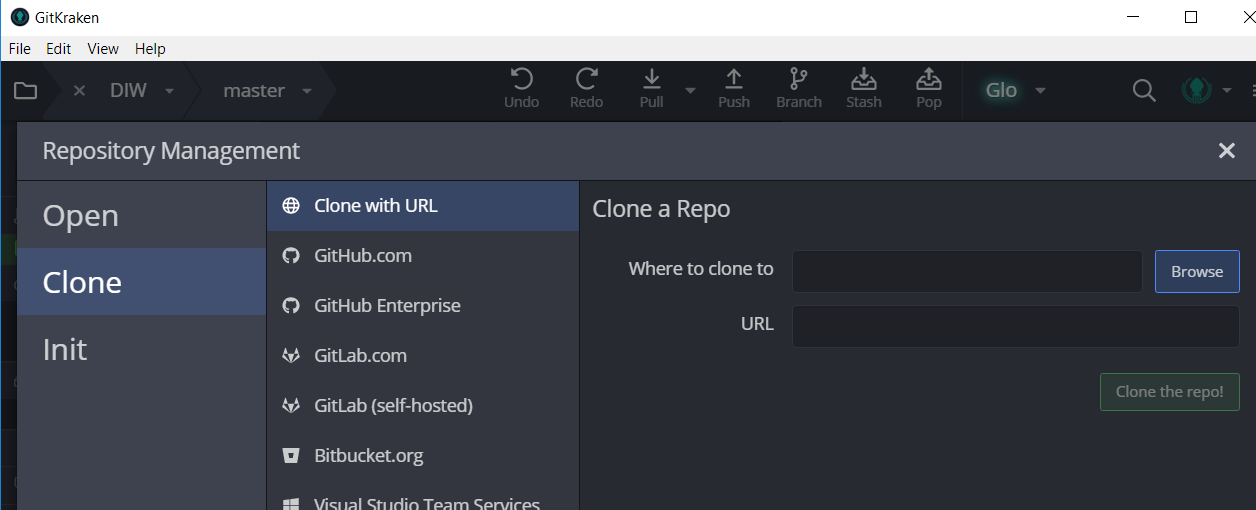
Desafío 2:

La instalación de GitKraken es similar a la de Git, desde la página oficial cualquier usuario es capaz de realizar la instalación; usaremos nuestra cuenta ya creada de Git para tener acceso y almacenamiento en los repositorios.

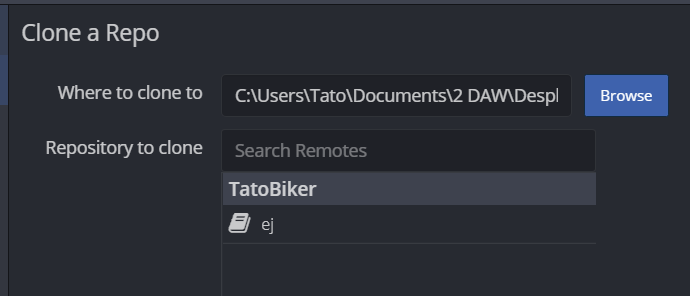
Al usar ya este programa en otra asignatura, tenemos conocimientos básicos de la aplicación y documentos almacenados. Mi cuenta en este caso:

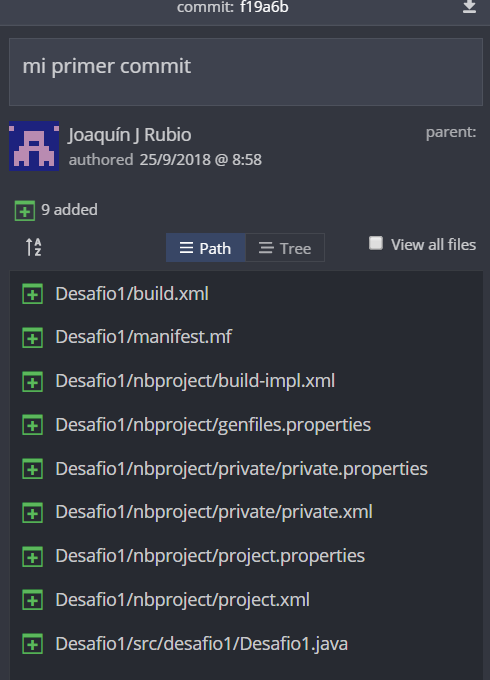


Para clonar el repositorio que teníamos creado: **File > Clone Repo** (arriba izquierda)

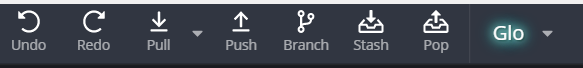


Debe aparecer una ventana como esta, donde seleccionamos desde donde queremos clonarlo donde deseamos almacenarlo en local. Al seleccionar GitHub me aparece ya:

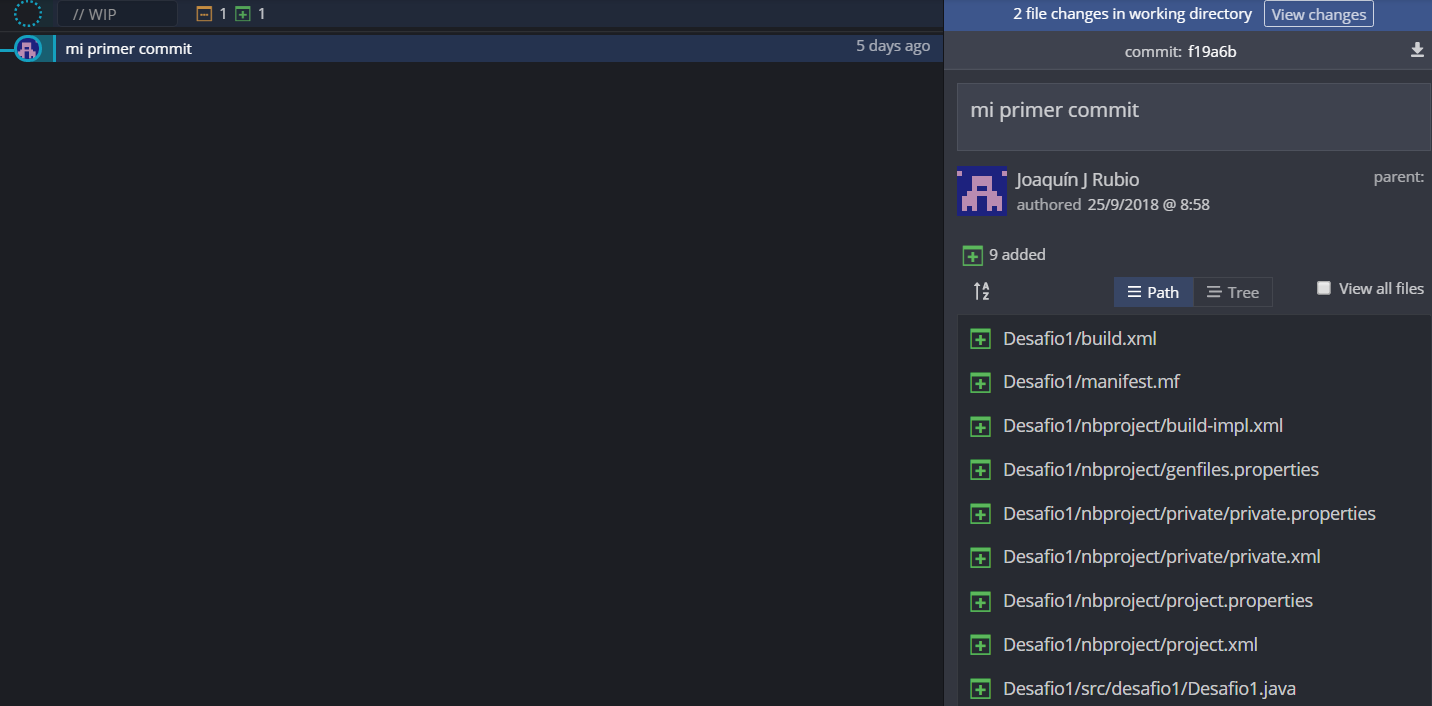


Ya tenemos el archivo tanto abierto en GitKraken como almacenado en la carpeta seleccionada:

Los principales comandos de Git, en GitKraken se muestran en esta barra de menú:

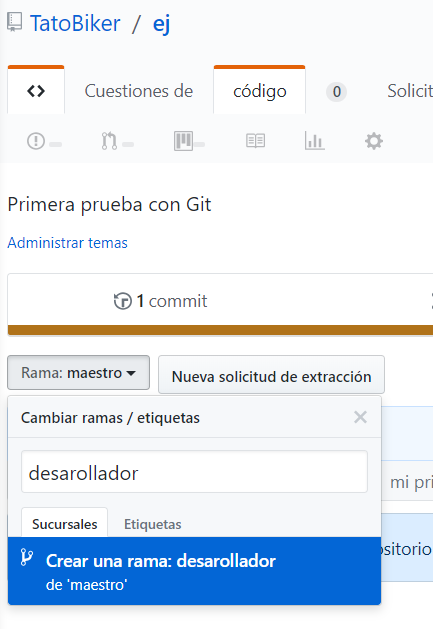


He modificado el archivo que acabo de clonar en el repositorio, y GitKraken te advierte de ello:

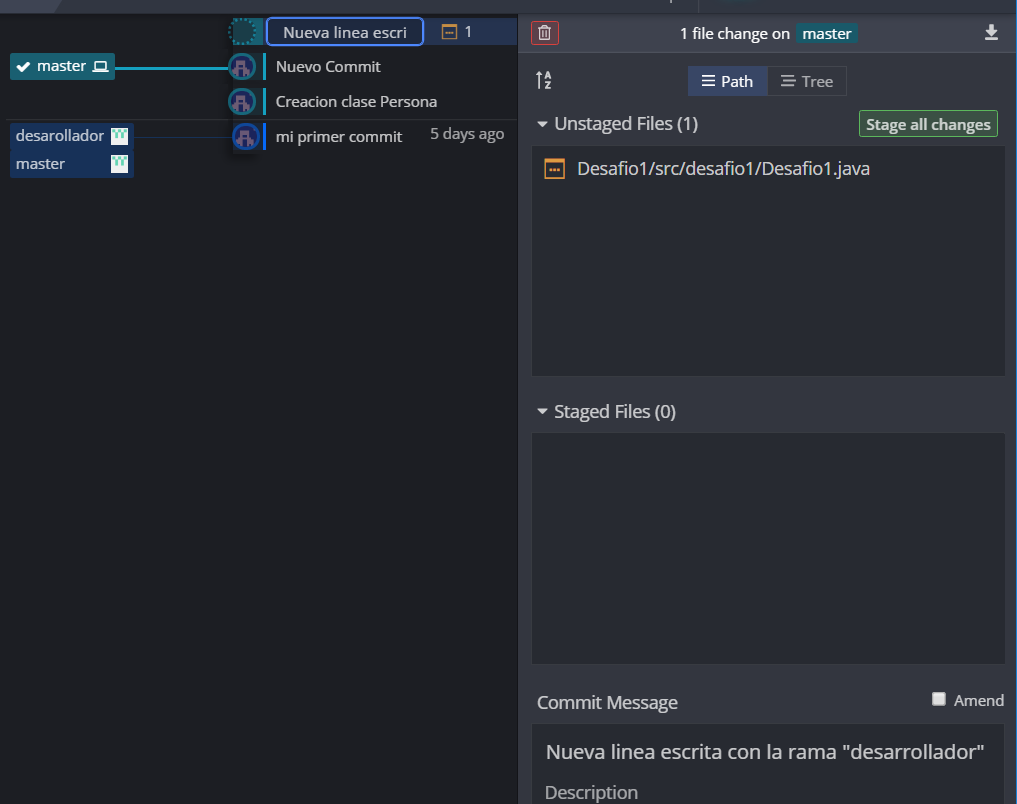


Le damos un nuevo nombre al **commit** y seguidamente al botón de **Push** para subirlo a GitHub. Con **pull**, tendremos la última versión guardada en el repositorio.

Para crear una nueva rama, lo hacemos desde GitHub:



Cuando se actualice GitKraken aparecerá “desarrollador”; si modificamos el archivo nos aparecerá algo similar, realizamos el commit y push después:



Para realizar un **merge**/pull request:

