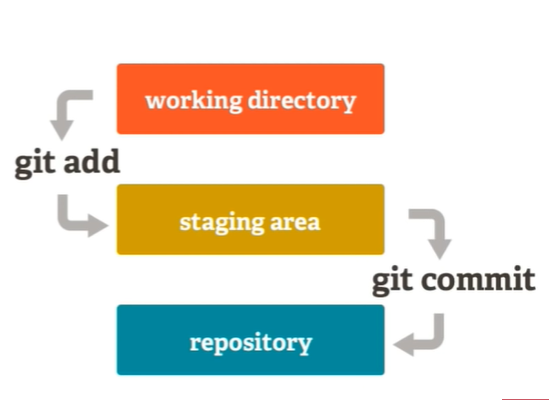
**PRIMEROS PASOS CON GITHUB:**

**03/01/2020**

1. **¿Qué es GIT?:**

* Controlador de versiones: Gestiona las distintas versiones de un proyecto.
* OpenSource.
* Asegura que siempre puedas tener la ultima versión del proyecto (que no este desactualizado el proyecto).
* Repositorio: Lugar donde vas a guardar el código.
* Commit: Git funciona a través de commits, son como snapshots (captura) de tu código en un momento concreto.
* En GIT cada archivo tiene uno de los siguientes estados(lugar donde esta):



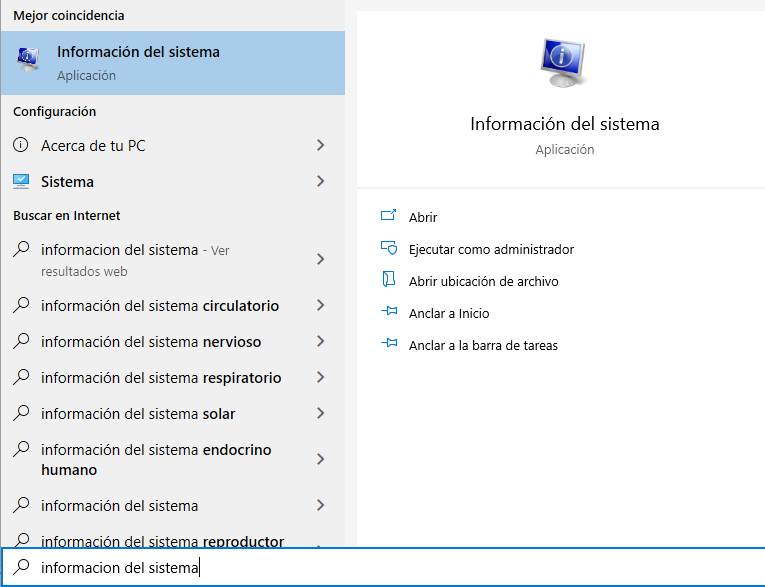
Working directory: Carpeta donde tenemos el código.

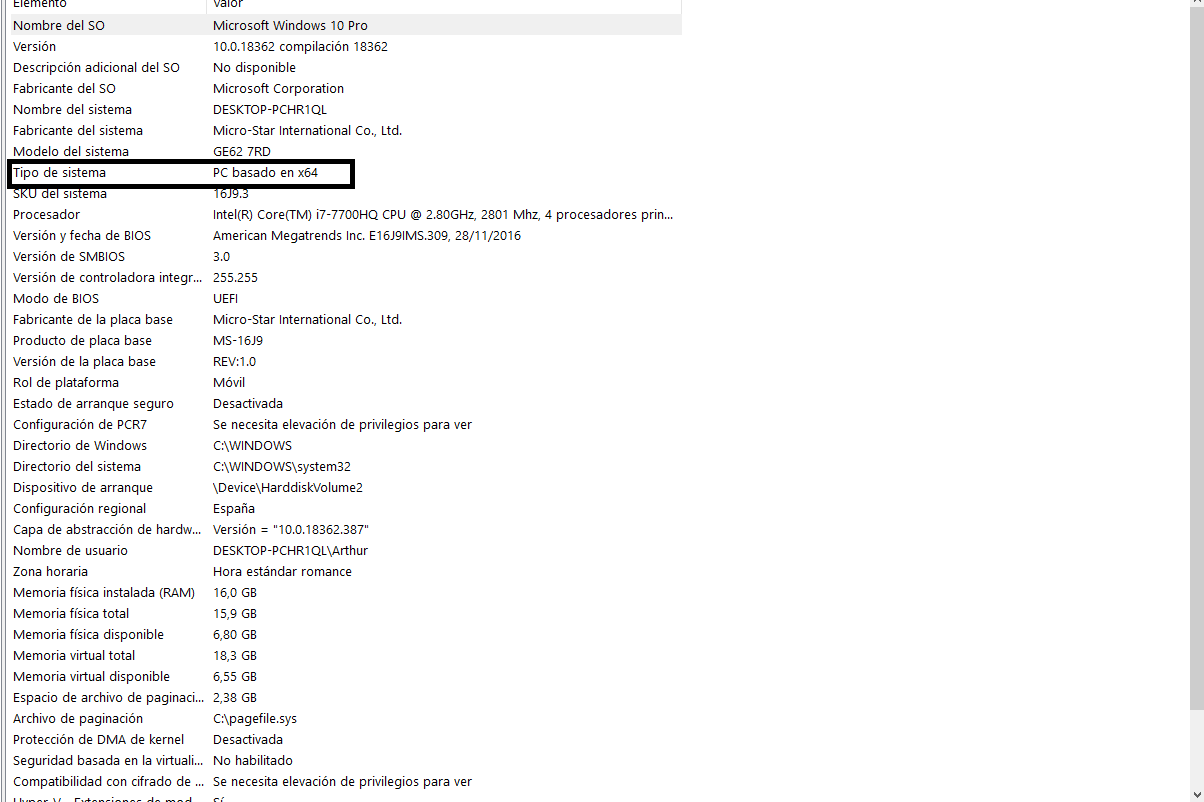
Staging área: Cambios preparados para añadirse al repositorio.

Repositorio: Cambios que ya has guardado.

1. **Comandos:**
2. git init: Se hace en la carpeta local donde tenemos el proyecto para indicar que vamos a usar GIT en ese proyecto.
3. git add <file>: Agrega file a la staging área.
4. git status: Ver en que estado están los archivos (working directory, staging área, repository).
5. git commit: Guardar los cambios del staging área al repositorio (local).
6. git push: Guardar los cambios del repositorio local en el repositorio remoto(GitHub, BitBucket…).
7. git pull: “Descargar” los cambios del repositorio remoto al working directory.
8. git clone: Copia desde el repositorio remoto al working directory.
9. **Instalacion de GIT:**

* GIT: Herramienta que usaremos para gestionar todos los archivos locales con el repositorio online.
* Instalacion:
* Descargamos GIT de la pagina oficial: <https://git-scm.com/downloads>
* Seleccionamos la descarga para el SO que tengamos.
* Seleccionamos el de 64 bits o 32 en función de la versión de SO que tengamos. Como mirarlo en Windows:



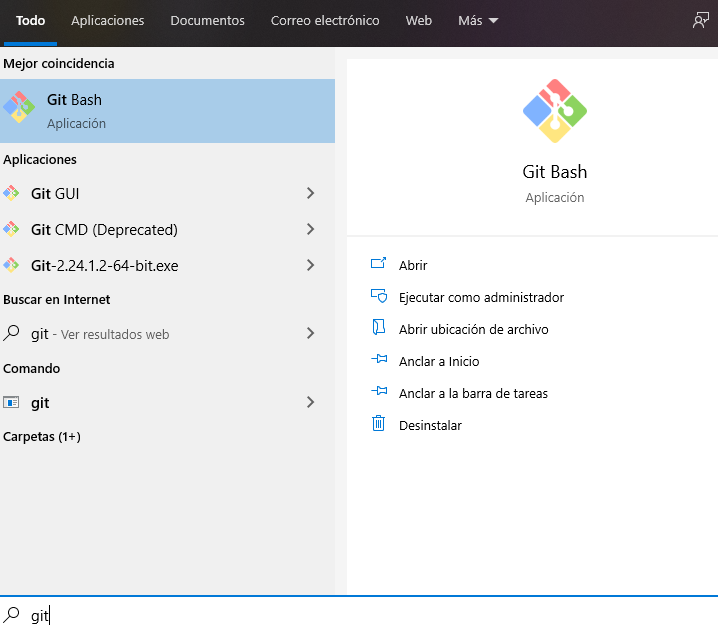


* En las opción “adjust your PATH environment” seleccionamos la segunda: “Git from the command line and also from 3rd-party software”.
* En el resto de opciones seleccionamos las siguientes:

1. Use the OpenSSL Library.
2. Checkout Windows-style, commit Unix-style line endings.
3. Use Windows default console window.
4. Enable file system caching, enable Git Credential Manager, enable symbolic links.
5. Not enable experimental, built-in add –i/-p.

* Pinchamos en Install.

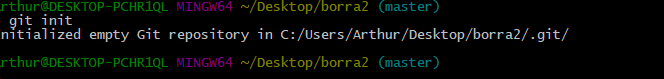
1. **Uso de Git:**
2. **¿Como subir nuestro codigo al repositorio de Github?:**
3. Abrimos GIT bash (interprete de git por consola).



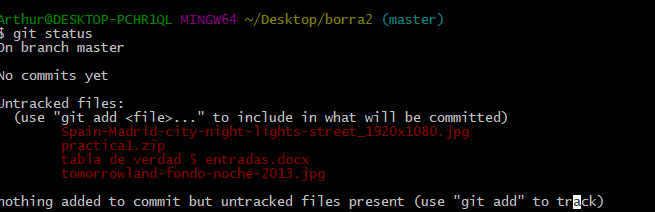
1. Navegamos desde la consola al proyecto que queremos subir: (Recuerda: cd: Acceder a una carpeta, dir: Ver lo que tiene esa carpeta; en Linux o Mac usaríamos cd y ls).



1. Inicializamos la carpeta para indicar que vamos a usar git en ella:

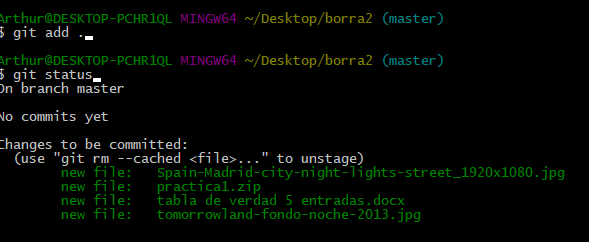


1. Comprobamos con git status que archivos hay sin subir al Staging Area (preparados para commit).



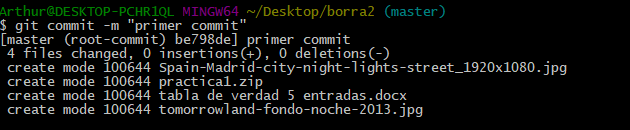
1. Vamos a subir todos los archivos en rojo al repositorio de GitHub.

* Usamos “git add <file>” para subir cada archivo o bien “git add .” para subir todos los archivos del directorio actual.

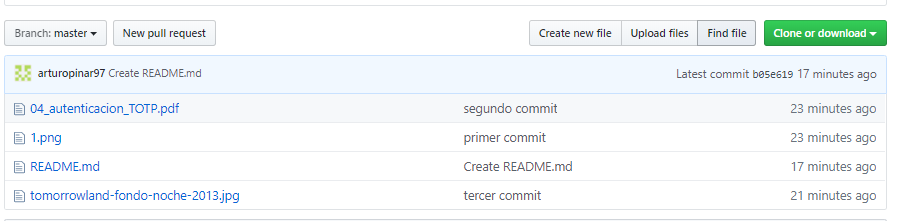


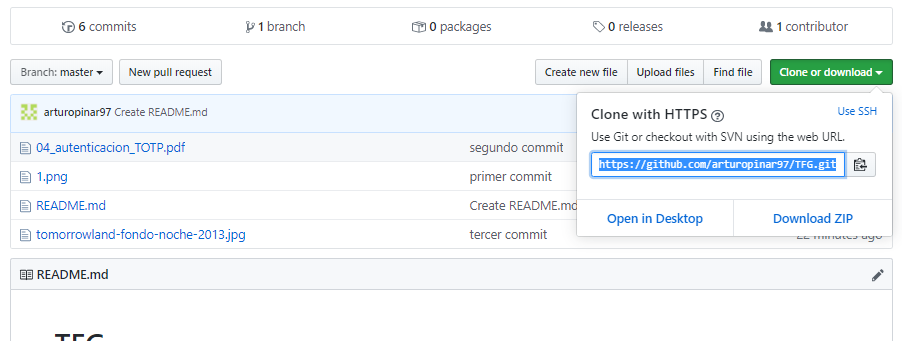
Todos los archivos en verde están preparados ya en el stagging área para hacer commit.

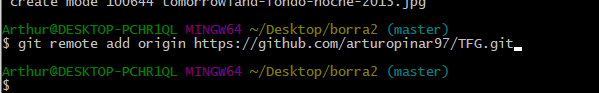
1. Subimos los archivos al repositorio **local** de git: Usamos git commit –m <mensaje del commit>:



1. Nuestro ultimo paso será subir los archivos al repositorio remoto (github) para ello primero tenemos que añadir la direccion HTTPs del repositorio a la variable “origin” de git (indica donde subir los cambios):

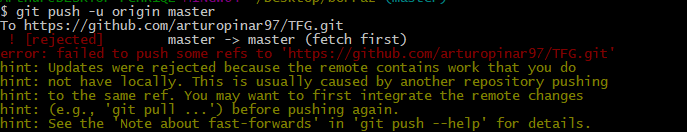






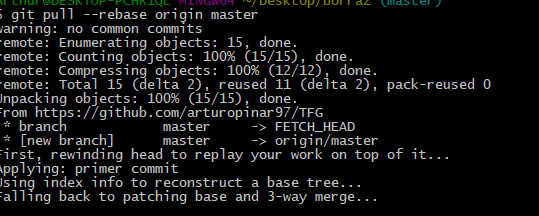
Ya tenemos la direccion del repositorio agregada. (Este paso solo hará falta hacerlo una vez para setear la variable ORIGIN).

1. Finalmente subimos el commit al repositorio a traves git push –u origin master (la primera vez escribimos ese texto que crea el enlace entre origin y master, luego ya (**siempre y cuando queramos subirlo a la misma rama ,s**ino en vez de master pondríamos el nombre de la rama) solo hará falta “git push”:

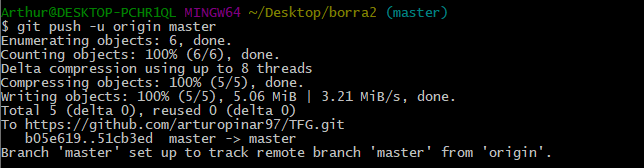


Es muy posible que nos aparezca el error que se ve en rojo, esto se debe a que si subimos estos cambios “machacamos” todo lo que ya había en el repositorio (**por eso la primera vez que hagamos push al repositorio hará falta hacer git pull para copiar los archivos que ya tiene el repositorio).**

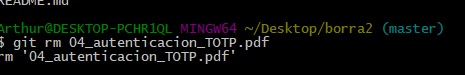
Por lo tanto solucionamos el error de la siguiente forma: [Referencias](https://stackoverflow.com/questions/24114676/git-error-failed-to-push-some-refs-to).



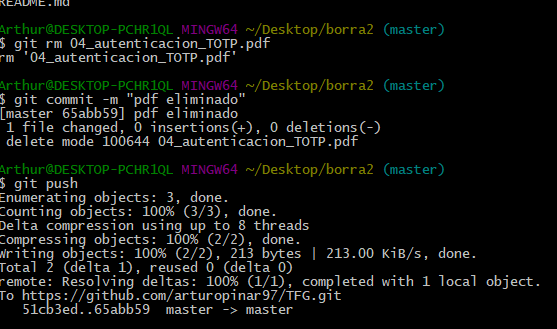
Ahora que ya esta actualizado ya podemos subir el commit con git push –u origin master.

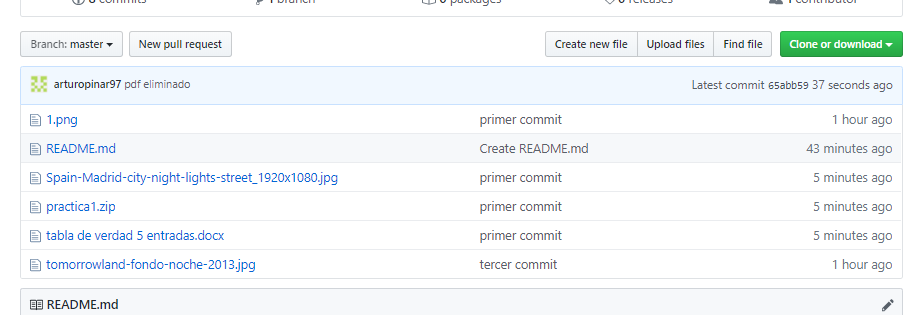


1. **¿Cómo eliminamos otros archivos del commit?:**
2. Una vez hemos subido los archivos usamos git rm <archivo> para eliminar del repositorio local el archivo.



1. Despues hacemos commit y push para subir el snapshot cambiado sin ese archivo a github.





Podemos ver que con el nuevo commit el pdf ya no esta en el repositorio.

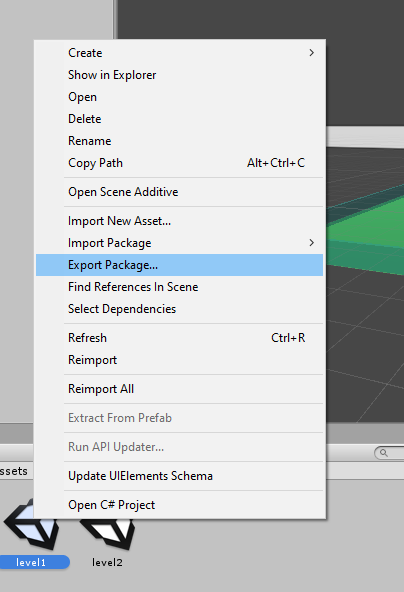
1. **Antes de hacer push:**

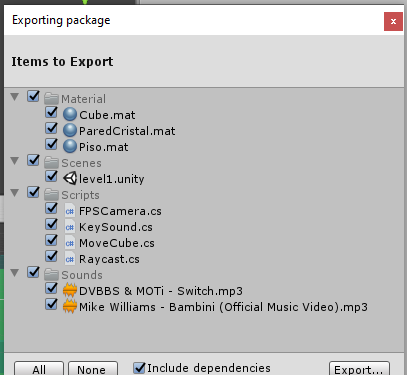
* Es de vital importancia no destruir el contenido existente en el repositorio de GitHub. Por ello, antes de subir nuestro commit (push) debemos hacer una copia de los archivos que había (pull). **De hecho se recomienda hacer pull siempre antes de commit para que el commit este actualizado).**
* IMPORTANTE: pull te agrega los archivos que no tienes en local del commit, es decir, que si tu eliminas un archivo en local aunque hagas git pull te va a decir que esta actualizado con master porque GitHub no ha detectado ningun cambio.
* POR LO TANTO ES MUY IMPORTANTE HACER GIT PULL ANTES DE PUSH PARA ACTUALIZAR EN LOCAL LA ULTIMA VERSION (COMMIT) SUBIDO DEL PROYECTO.

1. **Git Clone:**

* Git clone nos permite “clonar”, descargar el repositorio remoto como una carpeta (es decir lo usamos si queremos hacer una copia completa del repositorio), pull lo usamos en cambio para actualizar los archivos que nos falten de cualquier otro commit hecho por otro usuario.

1. **Como subir un proyecto de Unity a Github:**
2. Exportamos las escenas en formato Unity.





1. Realizamos todos los pasos anteriores en la carpeta donde lo hemos exportado para subirlo a Github:

git init.

git add .

git remote add origin https://github.com/arturopinar97/TFG.git

git pull –rebase origin master

git commit –m “primer cuatrimestre”

git push –u origin master