**Tarea 4.1**

**Programacion de juegos en grupo**

**Por José Manuel Atacho Rojas, Juan Manuel Cabello Cuerva y Alvaro Romeral Pineda**

**Índice**

[1. Parte de José Manuel Atacho Rojas 3](#__RefHeading___Toc177_2271711204)

[1.1. Primer Commit 3](#__RefHeading___Toc179_2271711204)

[1.2. Implementando las clases 5](#__RefHeading___Toc181_2271711204)

[1.3. Clase Final 7](#__RefHeading___Toc183_2271711204)

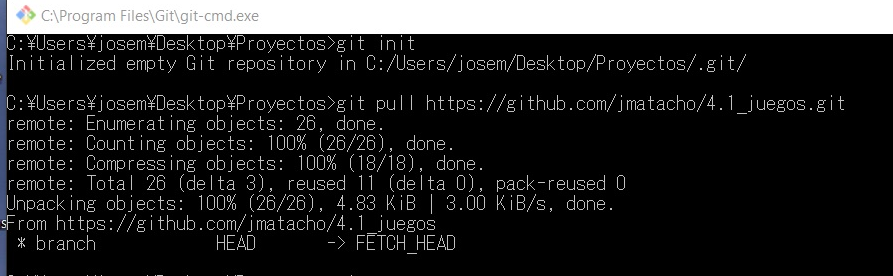
[1.4. Comprobar que tus compañeros pueden ver y actualizar el repositorio 8](#__RefHeading___Toc185_2271711204)

## 1. Parte de José Manuel Atacho Rojas

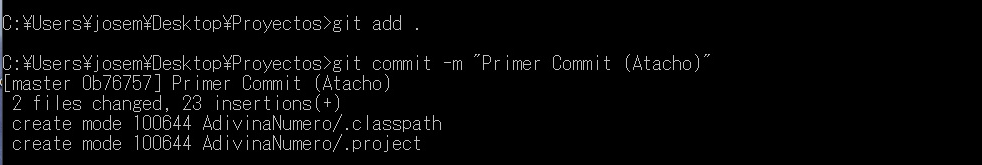
### 1.1. Primer Commit

*(Nota: En un principio la carpeta se llamaba AdivinaJuego, pero más tarde le cambie el nombre a Juegos)*

En nuestro primer commit, lo que tendremos que hacer es crear una carpeta, dentro de esta carpeta hacemos un **git init** para inicializar el proyecto, luego escribimos el comando “***git pull”*** seguido de la URL de nuestro repositorio creado anteriormente para que nuestra carpeta creada y el repositorio esten los dos sincronizados.



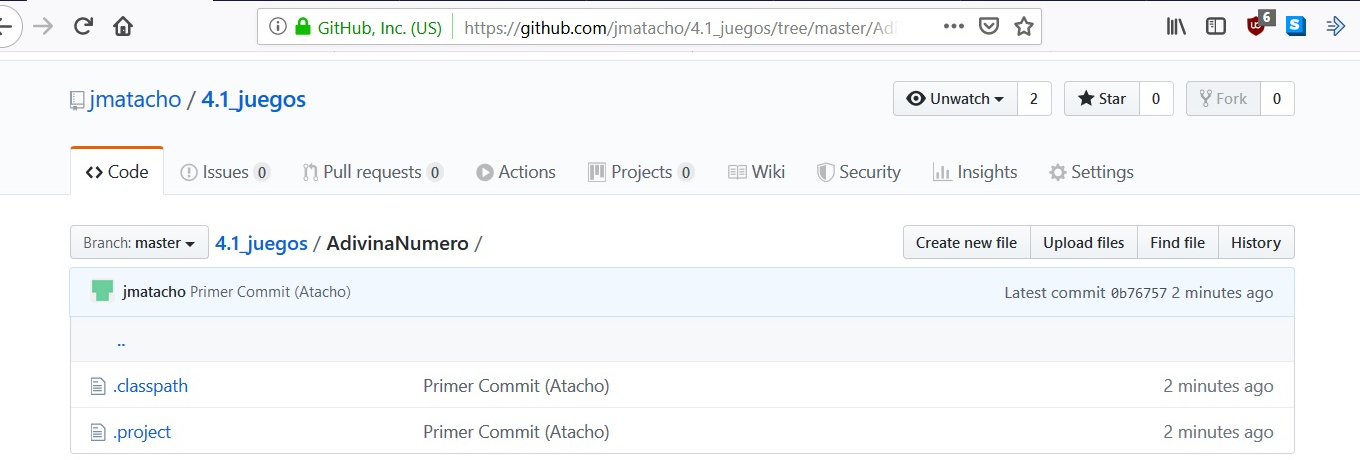
Luego, tras hacer el pull, haremos nuestro primer commit, en el cual subiremos la carpeta en la cual estarán nuestros juegos, esta de momento no tendra nada. Para hacer esto, primero tenemos que utilizar el comando “***git add .”*** el cual añadirá los cambios que hayamos hecho(en este caso crear una carpeta vacia), luego con el comando “***git commit -m***” escribimos el mensaje que tendrá nuestro commit.



Por ultimo utilizamos el comando “***git remote add***” para sincronizarnos con el repositorio que está en nuestro github y con el comando “***git push***” actualizamos nuestro repositorio.

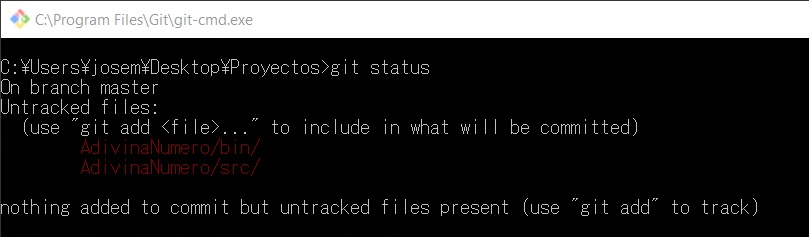


Para comprobar que los cambios se han realizado con éxito, nos dirigimos al repositorio a ver que los cambios, en efecto, se han hecho.



### 1.2. Implementando las clases

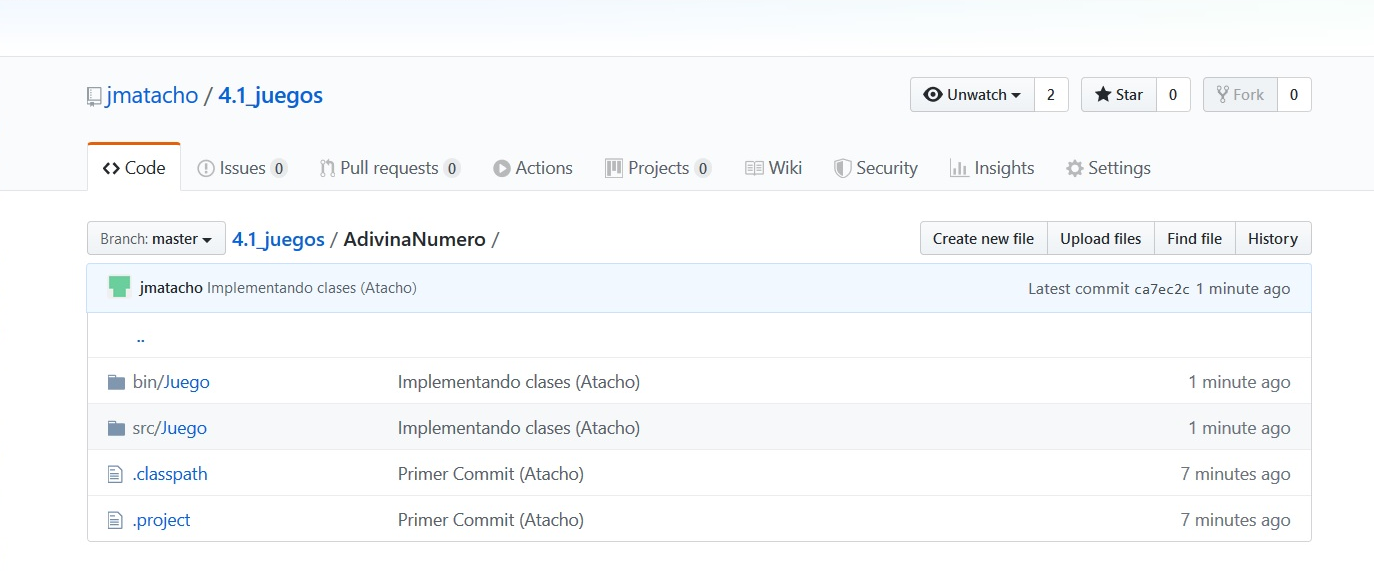
En esta parte, hay que actualizar el repositorio con el con el proyecto creado pero sin tener el juego hecho, simplemente una clase que este vacia. Para comprobar si un repositorio esta actualizado o no, simplemente utilizamos el comando “***git status***”. En este caso, como hemos actualizado el proyecto, este nos dirá que hay cosas que no estan subidas al repositorio.



Para actualizarlo hacemos lo mismo que hicimos anteriormente.

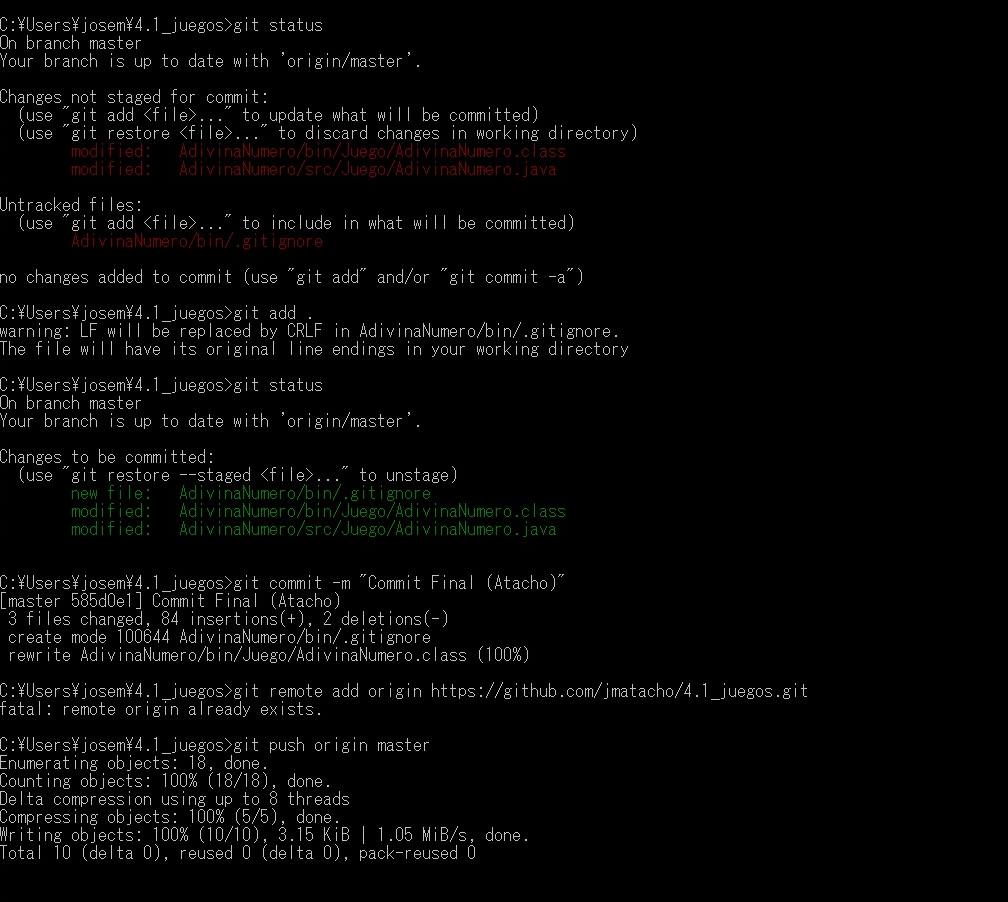


Tras hacer esto, volvemos al repositorio para ver que los cambios se han efectuado.

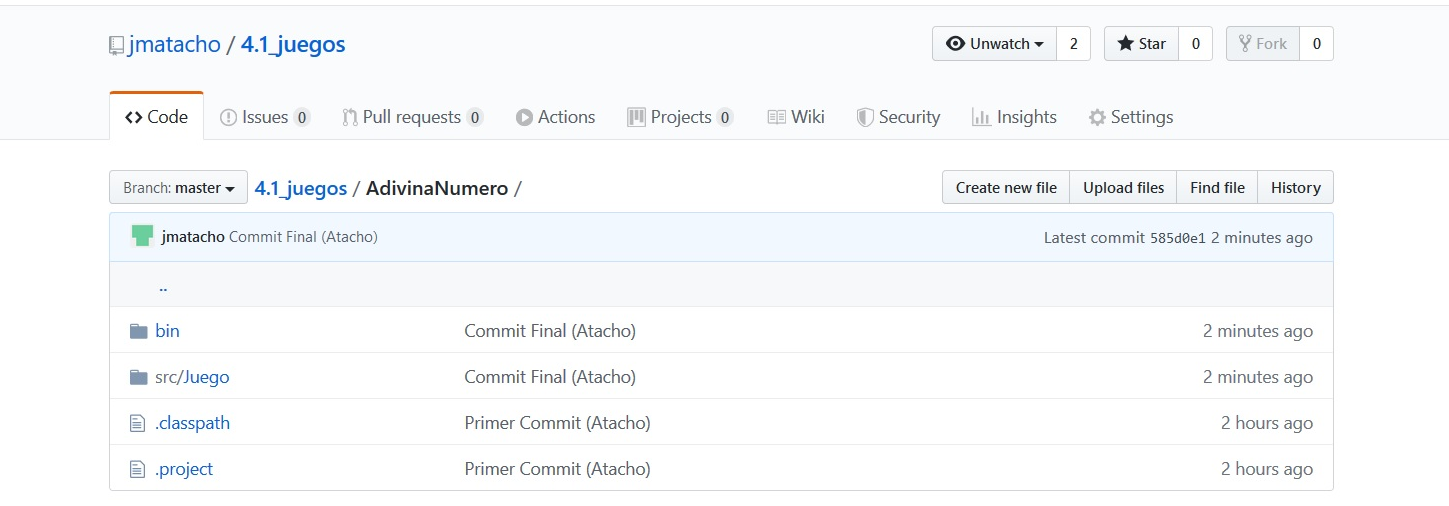


### 1.3. Clase Final

Una vez hecho el juego volvemos a hacer lo mismo que hemos hecho, utilizamos los mismos comandos para comprobar y actualizar el directorio.

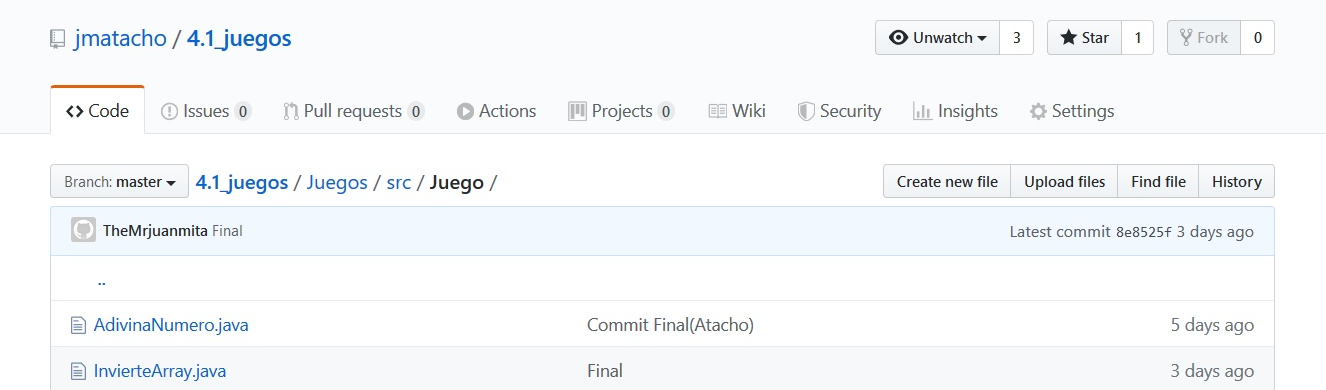


Ahora volvemos al repositorio para comprobar que los cambios se ha efectuado.



### 1.4. Comprobar que tus compañeros pueden ver y actualizar el repositorio

Cuando un compañero clone el repositorio y haga los cambios oportunos y lo vuelva a subir, podremos verlo sin problema alguno



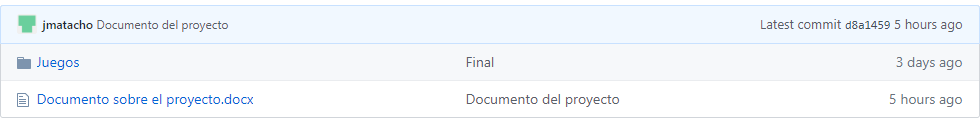
### 2.Juan Manuel Cabello

Implementación de la clase invierte array(Es el comit final ya que el primero ya estaba creado por atacho, al crear las carpetas en el repositorio)



### 2.1.Comprobar las actualizaciones de los demás

Aquí podemos ver como Atacho, es el último al añadir este documento al repositorio



Aqui comprobamos que ha sido subido correctamente y actualizado este archivo con nueva informacion

