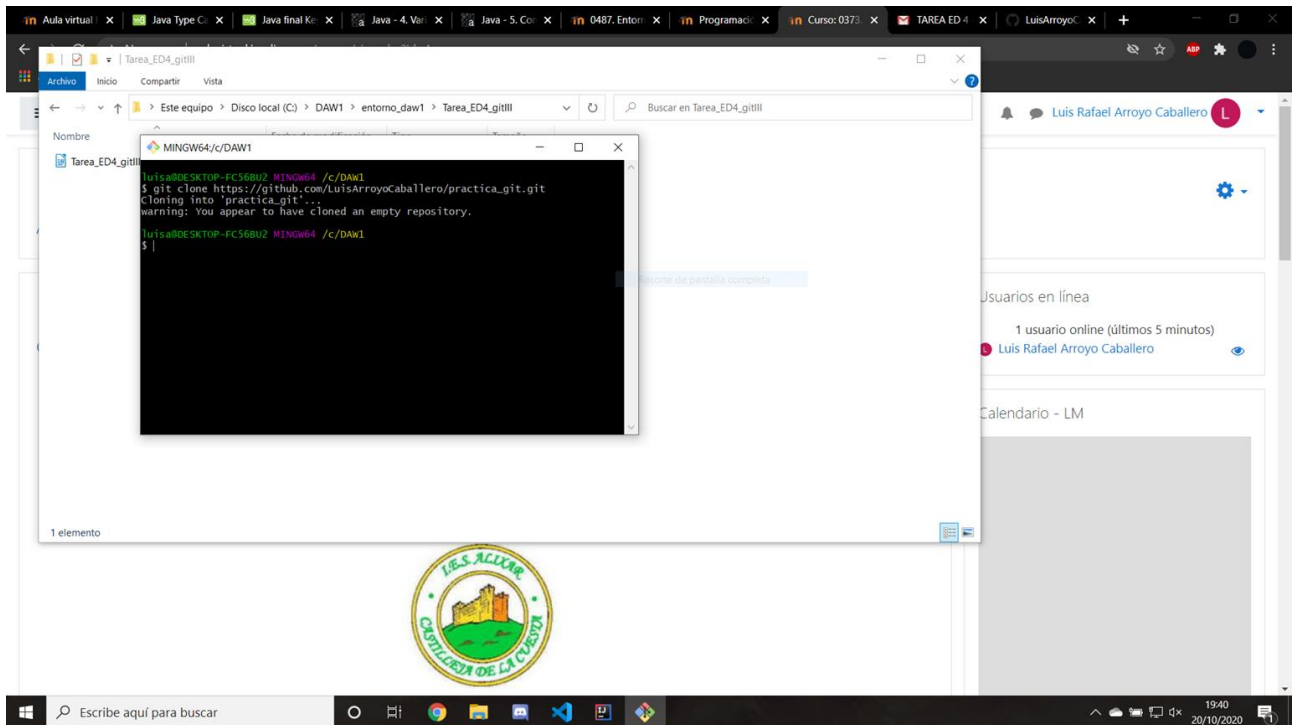


## Apartado 2: poniendo en práctica la herramienta Git

Con apoyo de la guía de supervivencia de git y github, contesta a los siguientes apartados. Ten en cuenta que, por cada uno, debes:

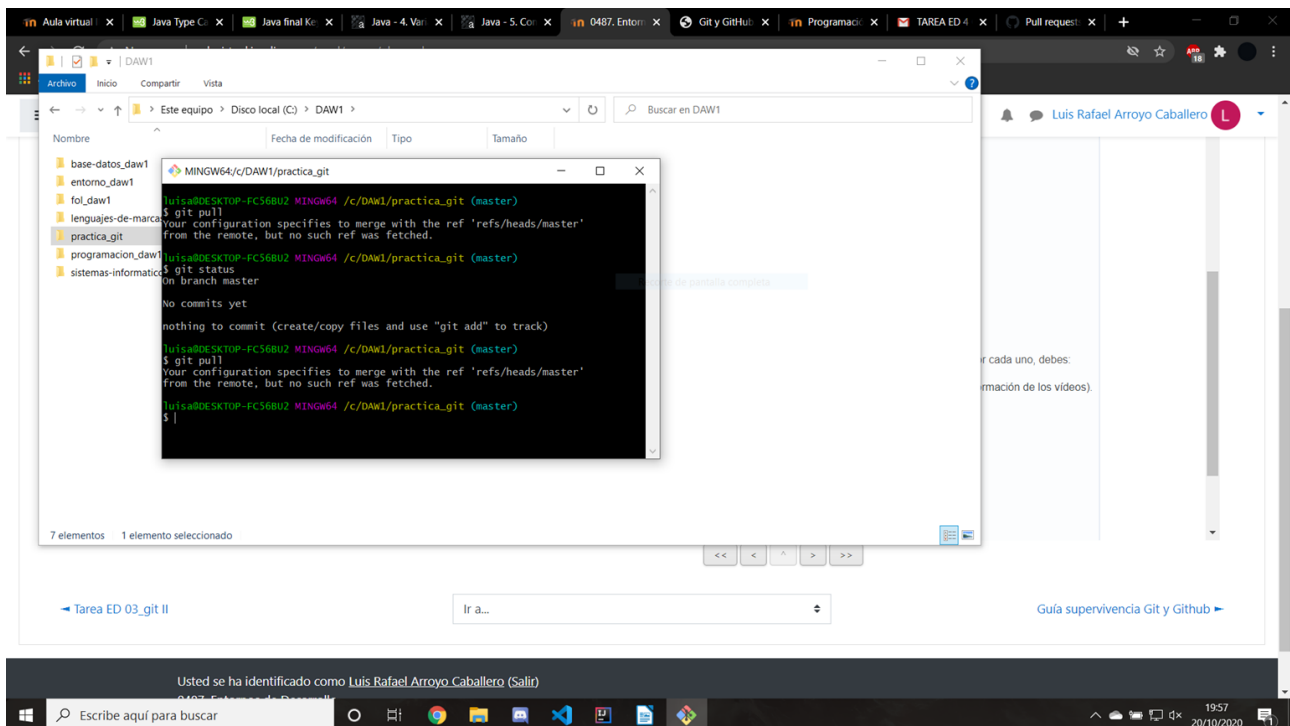
- Explicar en qué consiste y para que se usa (para esto, además de la guía, también te habrá sido útil la información de los vídeos).
- Seguir la práctica que indica la guía y aportar capturas que demuestren que lo han hecho.

### 1. Clonar un repositorio:



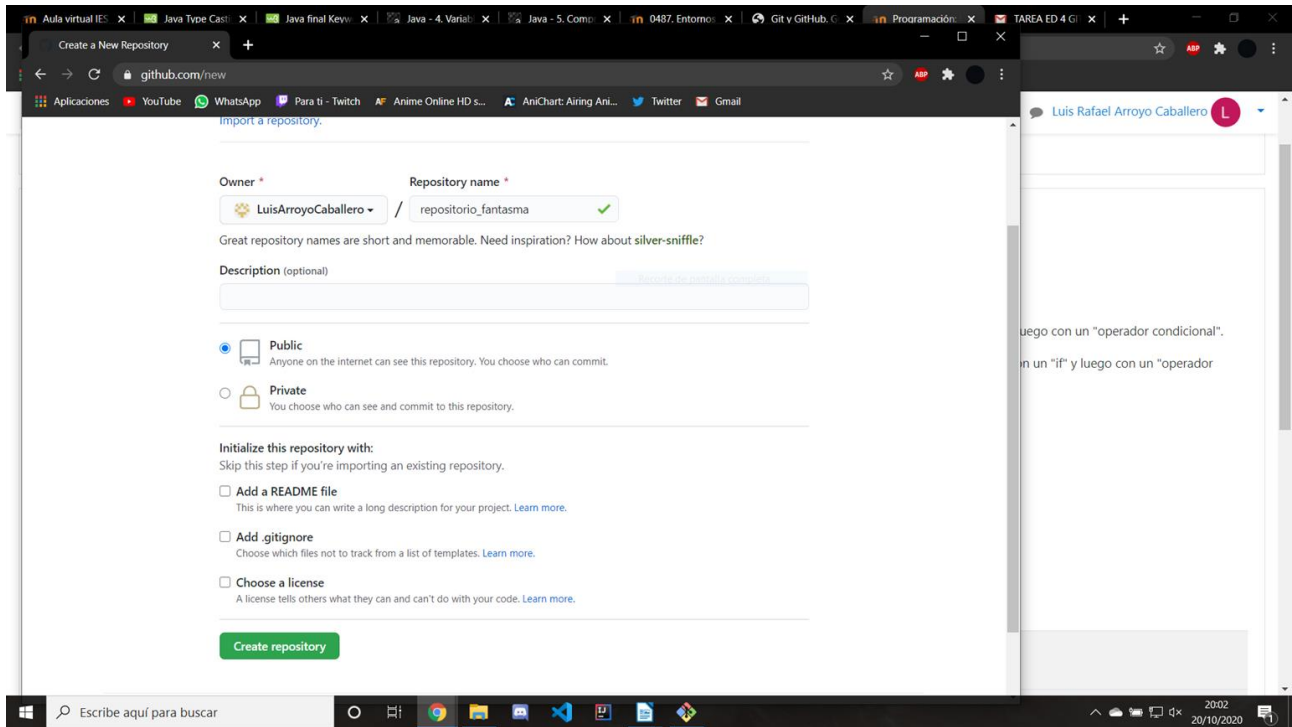
El comando **git clone “url\_repositorio\_github”**, sirve para copiar exactamente tu repositorio de github y pegarlo en tu máquina o equipo local.

## 2. Actualizar un repositorio de forma local.



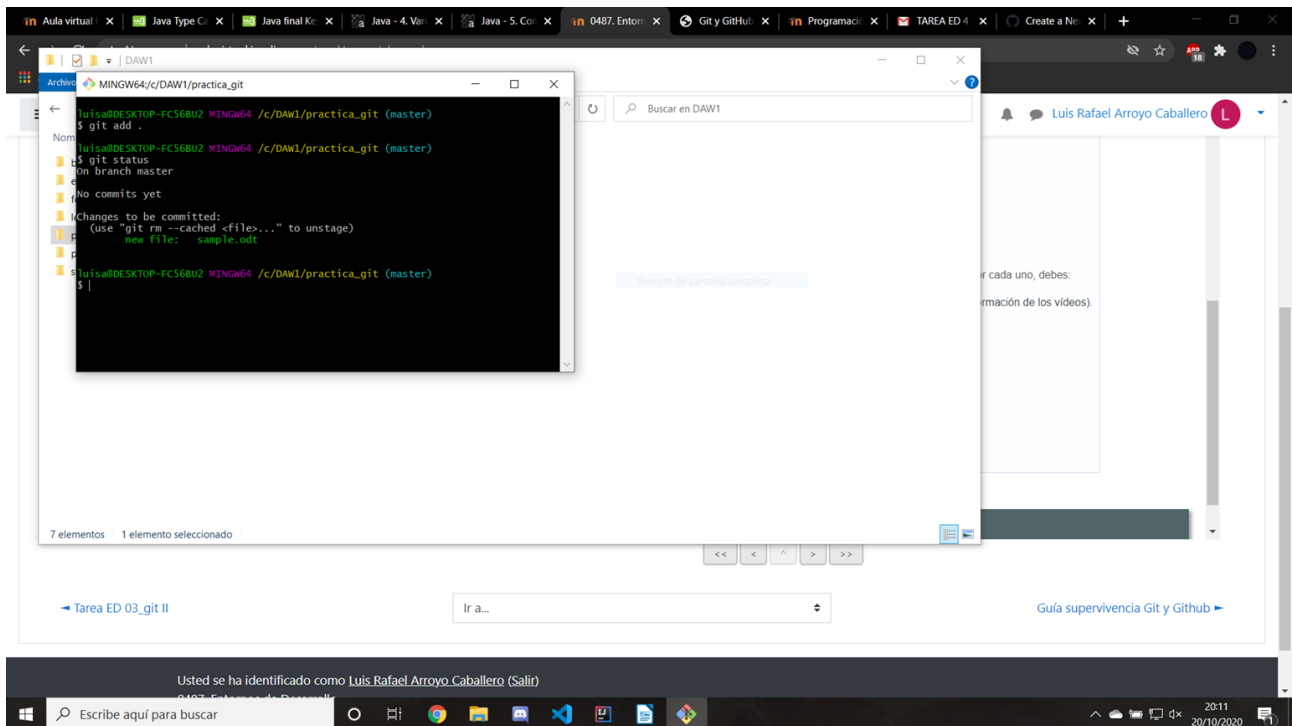
Para actualizar el repositorio de forma local usamos **git pull**, pero en mi caso al haber clonado un repositorio vacío no tenemos nada que actualizar. El comando **git pull** baja la información que tenemos en nuestra nube a nuestro equipo local, es decir, si otro usuario o desde otro equipo se actualiza el repositorio en la nube, nosotros “traeremos” esa información actualizada a nuestro equipo mediante este comando.

### 3. Crear un repositorio.

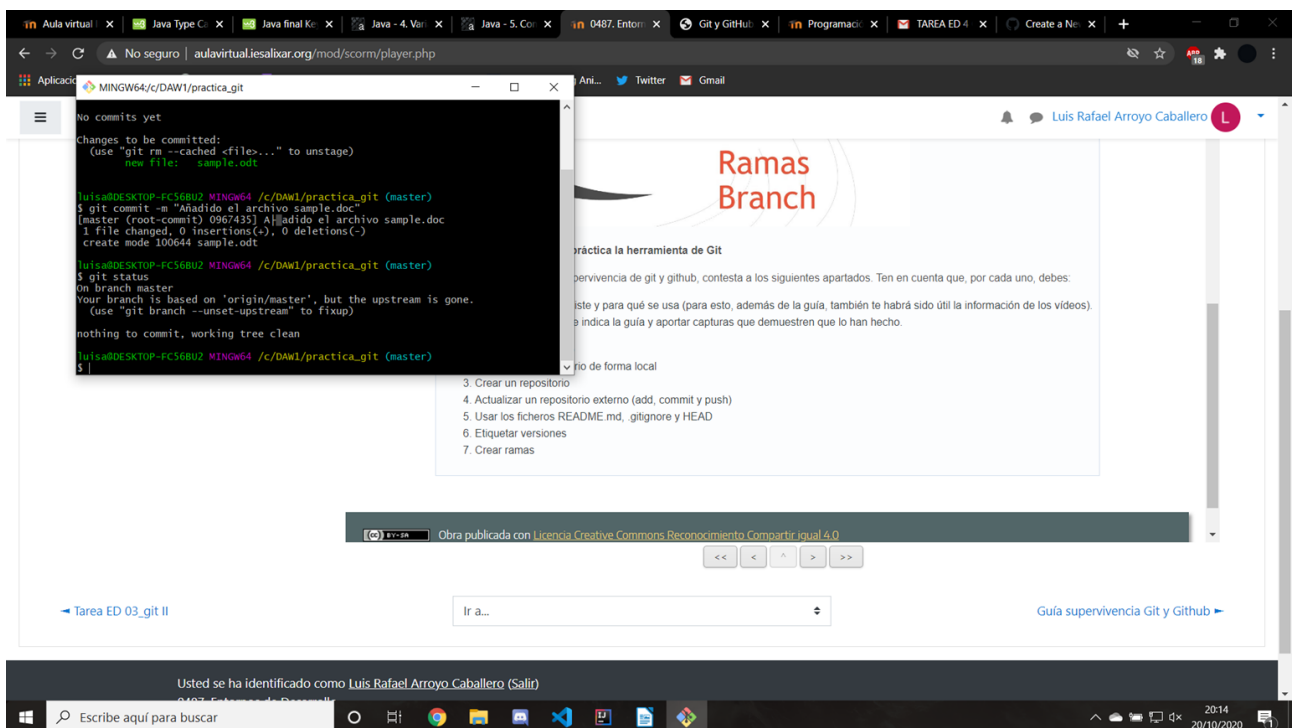


En este apartado no necesitamos usar ningún comando, simplemente accedemos a la página web de github y le damos al símbolo de “+” que tenemos arriba a la derecha → new repository → asignamos nombre al repositorio → pulsamos el botón “Create repository”. Una vez seguidos estos pasos tenemos nuestro repositorio creado, al clonarlo ya podremos subir nuestros archivos a la nube de github.

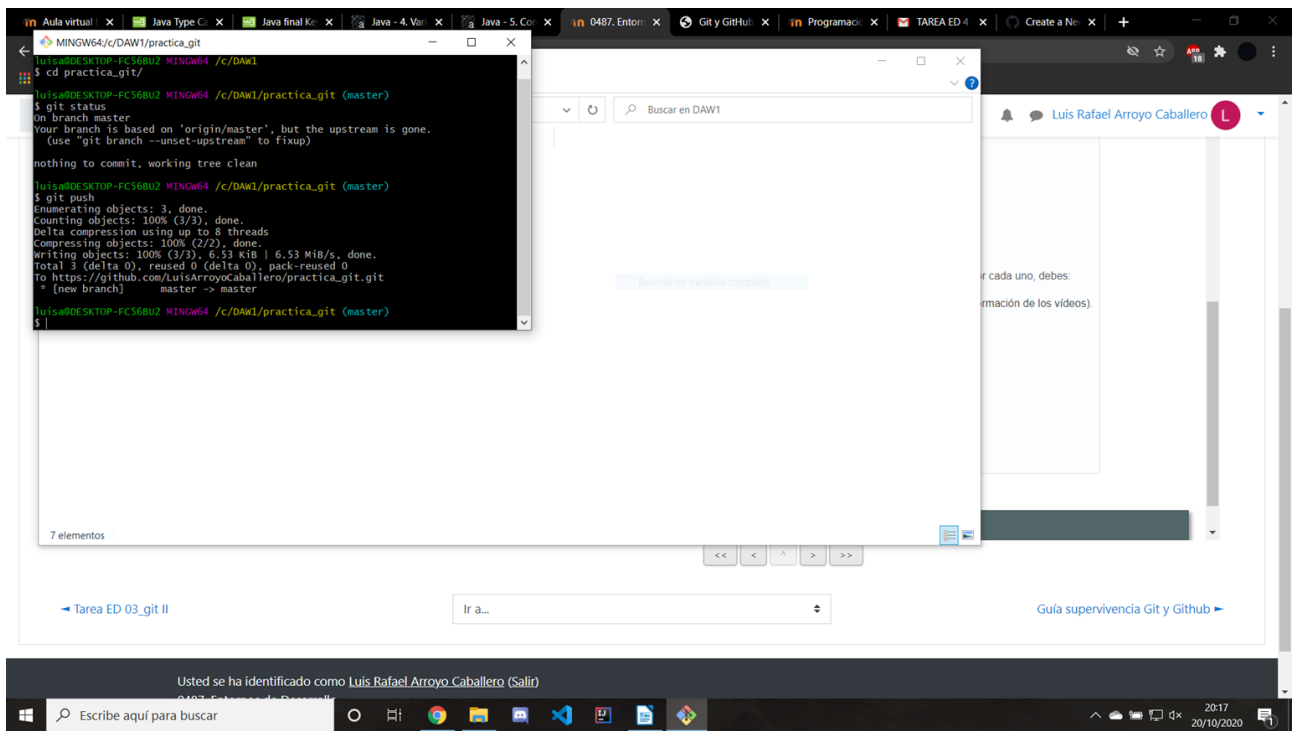
#### 4. Actualizar un repositorio externo (add, commit, y push).



- Hemos usado el comando **“git add .”**, que nos permite añadir la información que no sigue git al staging area (área de ensayo).



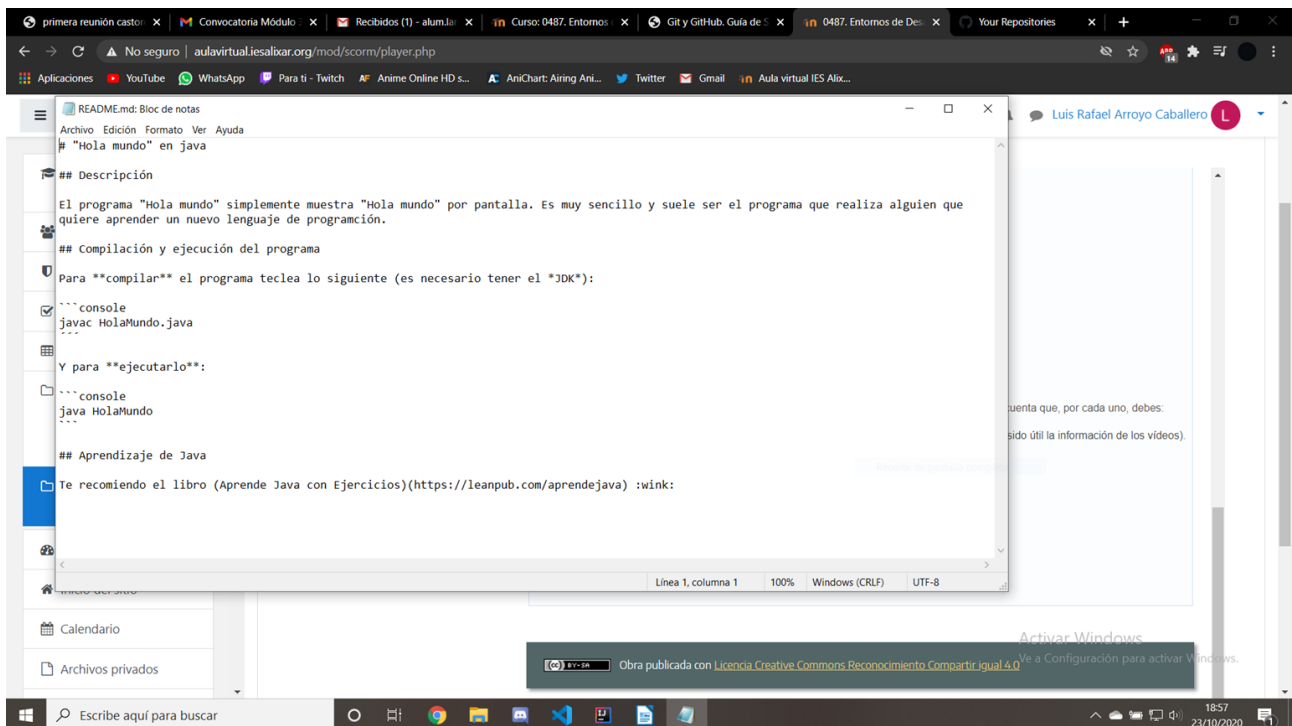
- Ahora utilizamos el **git commit -m “texto explicativo”** para confirmar los cambios hechos y que git le haga una instantánea (hace un nuevo “checkpoint”).



- Por último realizamos un **git push** para actualizar el repositorio en github, es decir, añadir los commits realizados entre el anterior push y este (en nuestro caso es el primer push).

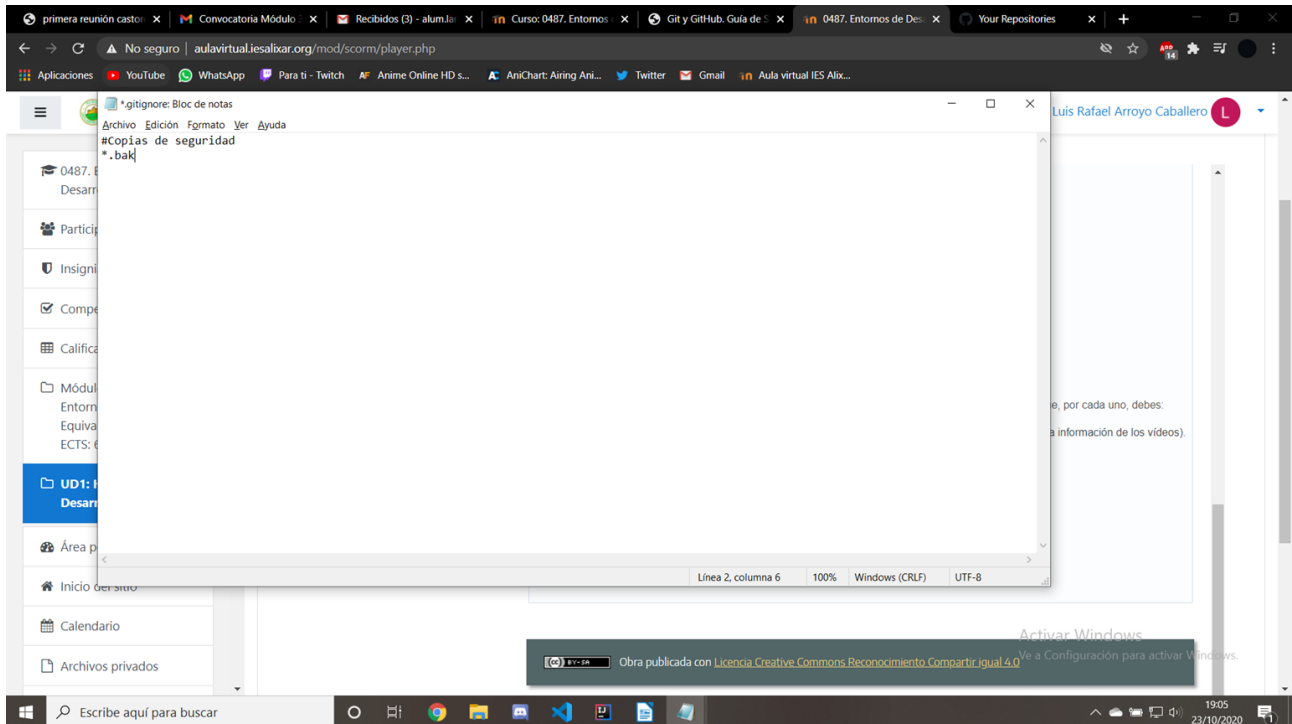
## 5. Usar los ficheros README.md, .gitignore y HEAD.

### README.md



He creado y modificado el archivo README.md en mi repositorio local, y ahí he introducido información sobre un programa sencillo para iniciar tu aprendizaje en java, una descripción, la compilación y ejecución del programa, la consola y una URL de una página con información para aprender java.

## .gitignore

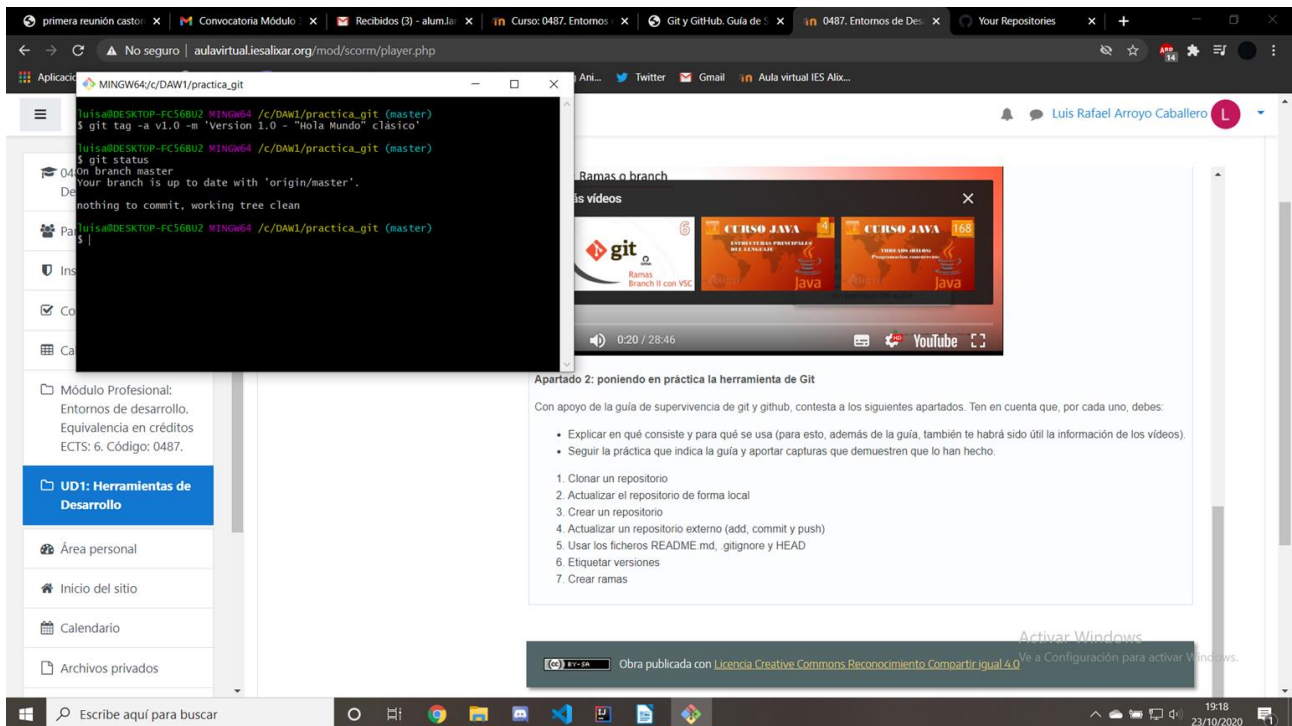


Creamos el archivo de texto .gitignore, e introducimos que extensión de archivo queremos que git no registre, es decir, que no le haga un seguimiento. Esto nos puede servir para ignorar copias de seguridad o compilaciones, pues solo queremos tener el código fuente en nuestro repositorio de github.

## HEAD

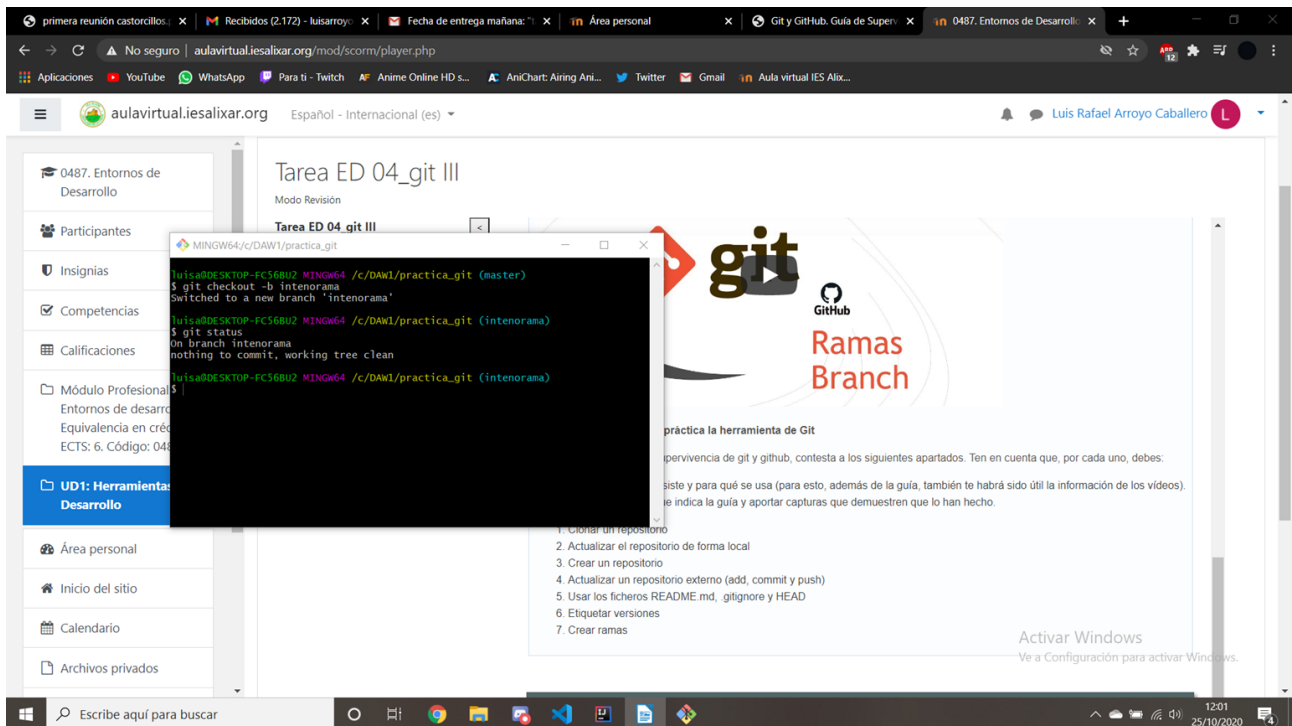
Es un archivo que te muestra el registro de todos los cambios que hemos hecho en el repositorio. Se puede mostrar de muchas maneras y podemos acceder mediante el comando git log.

## 6. Etiquetar versiones.



En esta imagen podemos ver como le hemos puesto una etiqueta con el comando `git tag -a`, este menos `-a` quiere decir que vamos a ponerle una anotación a esta etiqueta (en nuestro caso `v1.0`) y el `-m`, al igual que con el commit, vamos a dar un mensaje más detallado sobre esta etiqueta.

## 7. Crear ramas



Con el comando `git branch` creamos una rama y con el comando `git checkout` saltamos de una rama a otra. En este caso hemos usado un comando que realiza las dos acciones, que es **`git checkout -b intentorama`**. Con este comando hemos creado la rama `intentorama` y hemos saltado directamente a ella (como vemos en la terminal cuando hacemos `git status`). Puesto que ahora nos encontramos en la rama `intentorama` debemos indicar que estamos actualizando desde la rama, esto lo haríamos con el comando **`git push --set-upstream origin intentorama`** (este comando lo utilizaríamos después de añadir a la staging area los cambios y hacer un commit). La próxima vez que actualicemos nuestro repositorio, Git recordará la rama y no tendremos que especificarla de nuevo.

En caso de que queramos fusionar la rama `master` e `intentorama` debemos usar el comando `git merge intentorama` (nombre de la rama) + un `git push` para actualizar nuestro repositorio de github.