Práctica de laboratorio

Práctica de estudio de Git y GitHub, tutorial y aprendizaje

NOMBRE ALUMNO: Jhon Salazar Varela

ISOP411\_Git\_GitHub-JSVarela22.docx

Índice

[Objetivo de la práctica 2](#_Toc95245841)

[Inventario de material necesario 2](#_Toc95245842)

[Ejecución, investigación y desarrollo 3](#_Toc95245843)

[Consideraciones finales y metacognición 4](#_Toc95245844)

# Objetivo de la práctica

En esta práctica trataremos una herramienta muy utilizada en el ámbito de la informática para la compartición de archivos y SO libres o de pagos, y además un sistema de control del que dependen grandes versiones de Software libre del mundo en la actualidad, veremos sus historias, opciones, usos y entenderemos sus relaciones.

Con esta práctica se buscará crear un tutorial de inicio para tener conocimientos sobre el uso de estas herramientas y el como empezar a crear contenido para la plataforma GitHub, repositorios de código y demás cosas que permiten a la comunidad estar al corriente de muchos de los avances en errores o nuevos ámbitos informáticos.

# Inventario de material necesario

Para esta práctica necesitaremos lo siguiente:

* Un entorno físico donde almacenar la Máquina Virtual
* Entorno de Virtualización (VMware)
* Conectividad a Internet
* Las ISO correspondientes de los SO que vamos a utilizar (Linux Ubuntu)
* Cuenta gratuita en GitHub
* Correo Electrónico
* Credenciales para Git y GitHub

# Ejecución, investigación y desarrollo

## Estudio de Git y GitHub

Comenzando por **GIT** se trata de un sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto.

Las características básicas de rendimiento de Git son muy sólidas en comparación con muchas otras alternativas. La confirmación de nuevos cambios, la ramificación, la fusión y la comparación de versiones anteriores se han optimizado en favor del rendimiento. Los algoritmos implementados en Git aprovechan el profundo conocimiento sobre los atributos comunes de los auténticos árboles de archivos de código fuente, cómo suelen modificarse con el paso del tiempo y cuáles son los patrones de acceso

Seguido de este tenemos por parte de Microsoft, **GitHub**, que se trata de una una plataforma de alojamiento, propiedad de Microsoft, que ofrece a los desarrolladores la posibilidad de crear repositorios de código y guardarlos en la nube de forma segura, usando un sistema de control de versiones, llamado Git.

GitHub está basada en el sistema de control de versiones distribuida de Git, por lo que se puede contar con sus funciones y herramientas, aunque GitHub ofrece varias opciones adicionales y su interfaz es mucho más fácil de manejar, por lo que no es absolutamente necesario que las personas que lo usan tengan un gran conocimiento técnico.

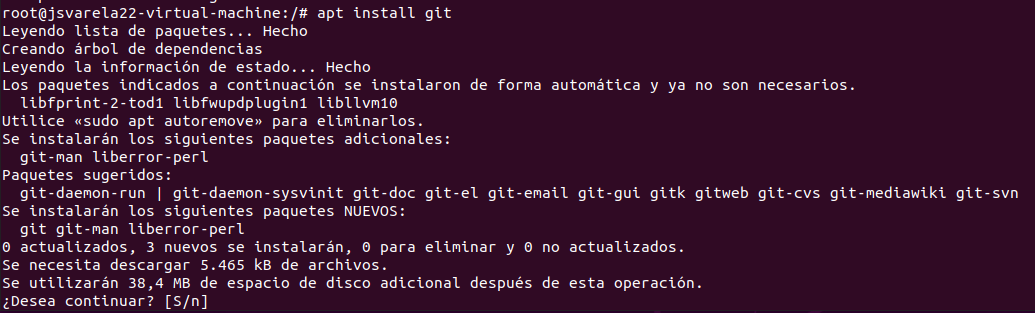
Las principales características de la plataforma es que ofrece las mejores características de este tipo de servicios sin perder la simplicidad, y es una de las más utilizadas del mundo por los desarrolladores. Es multiplataforma, y tiene multitud de interfaces de usuario.

Además de permitirte mirar el código y descargarte las diferentes versiones de una aplicación, la plataforma también hace las veces de red social conectando desarrolladores con usuarios para que estos puedan colaborar mejorando la aplicación.

## Relación

Para llegar a relacionar estas dos herramientas, debemos tener en cuenta que hace falta una creación previa de un correo electrónico en el que podamos validar nuestras credenciales y darnos de alta en las paginas oficiales de las plataformas.

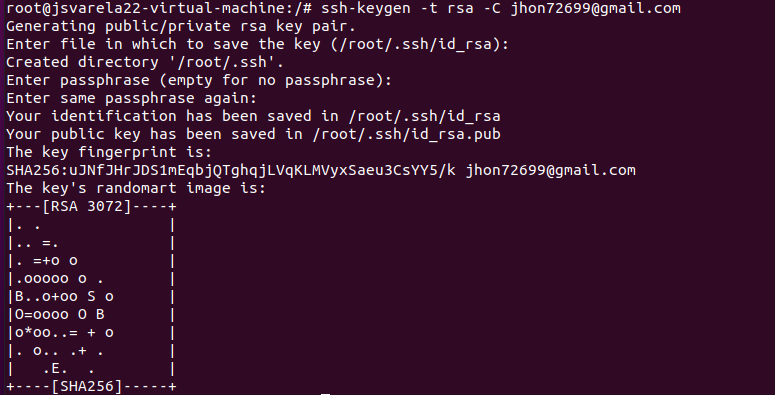
Seguido de esto, veremos como instalar git en un sistema operativo Linux

Teniendo un equipo, MV, con un sistema operativo funcional y una cuenta de usuario creada, entraremos en el terminal de Linux y nos tendremos que instalar la herramienta de Git, actualizaremos de primera mano la librería con “apt update” y seguido de esto aplicaremos el comando de instalación de git “apt install git”.

Al instalarse podremos ya empezar a relacionar nuestras dos herramientas con nuestro correo electrónico, utilizaremos en este caso uno personal con el que ya tengas una cuenta creada en GitHub previamente (aunque no es necesario, lo hago asi porque este es mi caso).

Seguiremos este comando para configurar por nuestro correo electrónico una vinculación.

Y con esta parte hecha veremos la opción para generar una ssh key de forma que nos de mas seguridad a la hora de conectarnos a la plataforma de GitHub.



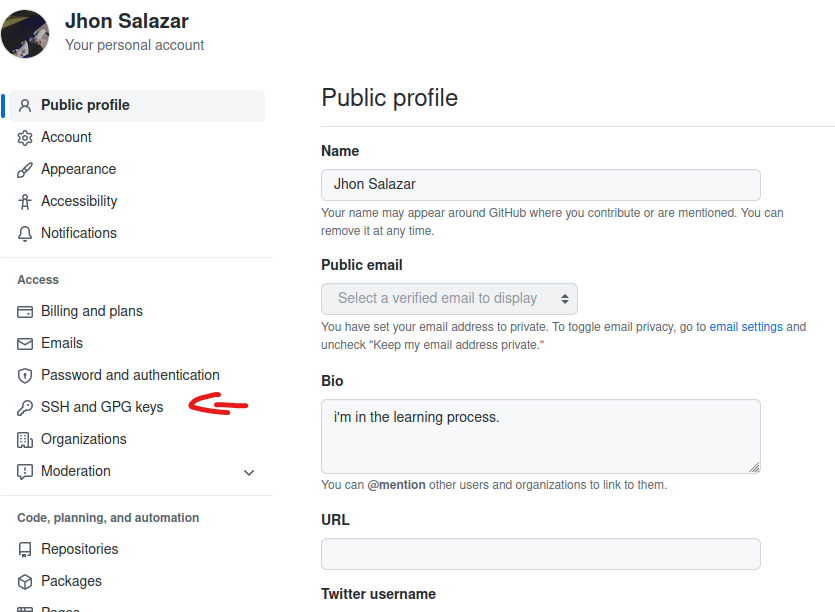
Esta ssh key nos servirá como ya hemos dicho para una mayor seguridad a la hora de conectarnos a la plataforma de GitHub, con lo que tendremos que guardarla y mantenerla en un lugar asegurada de que se nos pierda.

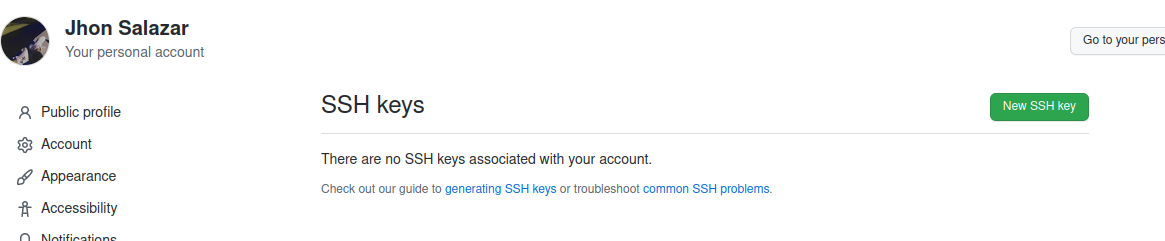
Seguido de esto buscaremos la manera de copiar nuestra SSH key para poder incluirla en nuestra cuenta de GitHub, para ello, nos instalaremos la siguiente herramienta “Xclip” con esta podremos copiar la key y pasarla a nuestra cuenta.

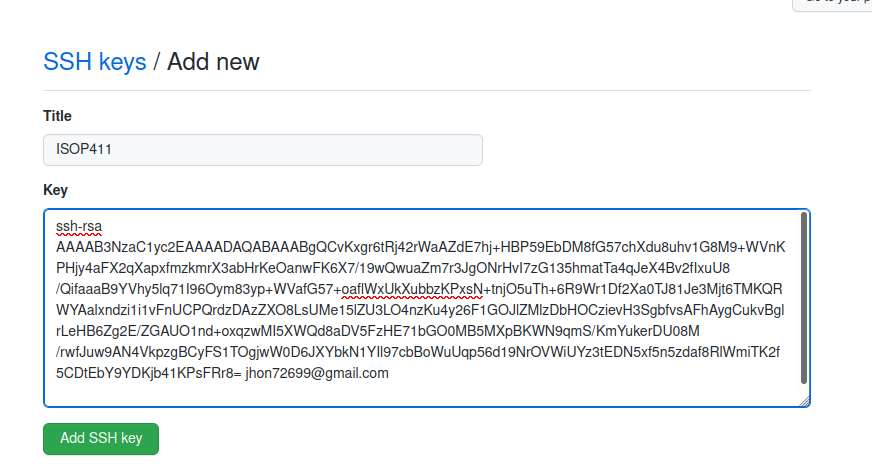
Seguiremos el siguiente comando de instalación “apt install xclip” y luego utilziaremos este comadno para copiuar la ssh key:

* “Xclip -sel clip <-/.ssh/id\_rsa.pub”

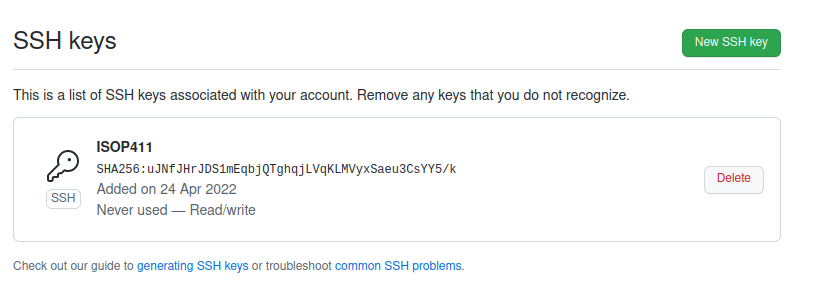


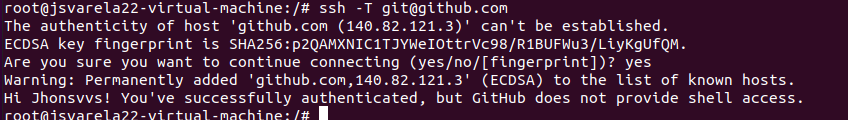
Ahora entraremos en nuestra cuenta de GitHub desde la pagina web, nos iremos a las configuraciones de nuestro perfil y en el apartado de SSH and GPQ KEYS seleccionaremos “SSH Key”



Al seleccionar esta opción dispondremos de un recuadro en el que podremos pegar lo copiado con la herramienta xclip, es decir, la ssh key que hemos generado acorde con nuestro correo electrónico vinculado a Git.

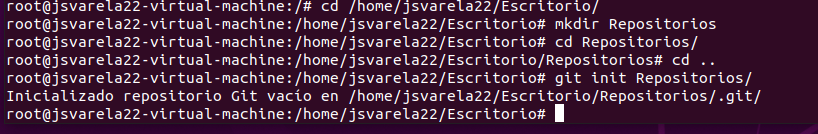
Se nos quedaría algo asi:

Aceptaremos y añadiremos la ssh key para tener ya nuestra cuenta asegurada.

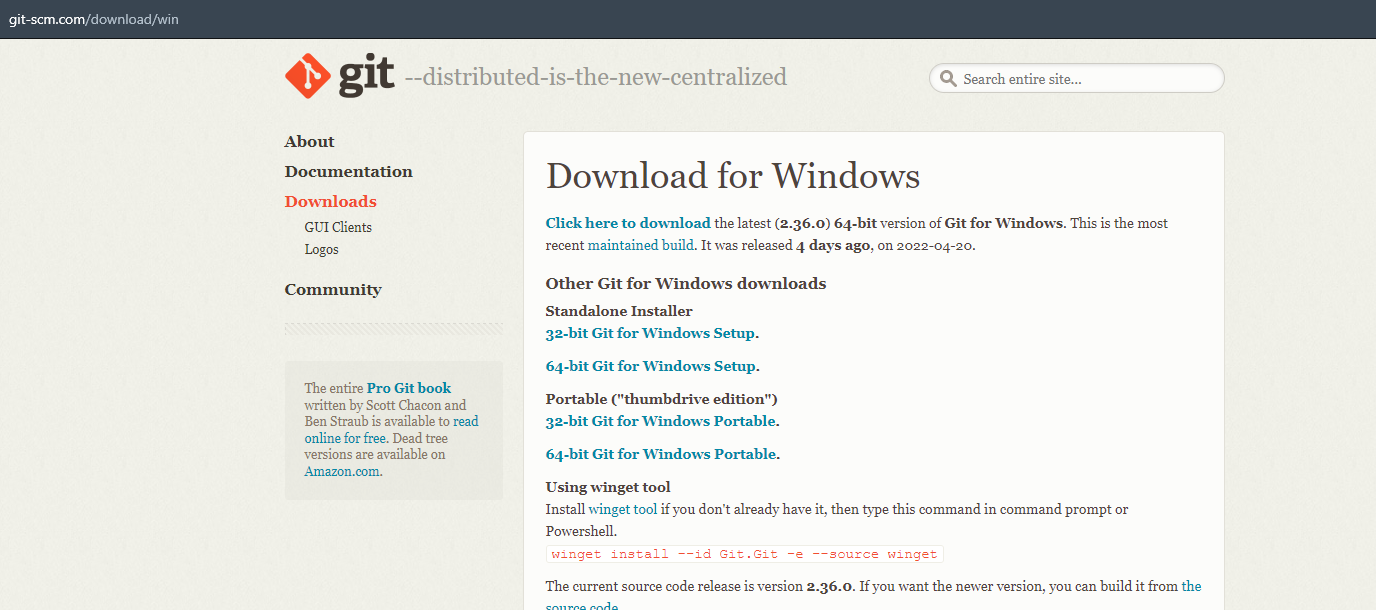
Una vez terminado esto comprobaremos la relación que habremos creado entre Git y Github de la siguiente manera, seguiremos el comando “ssh -T [git@gitHub.com](mailto:git@gitHub.com)” y aplicaremos las opciones que nos propone con “Yes”, esto nos autentificará en la plataforma de Github con Git y nos dará acceso.

## Creación de repositorios

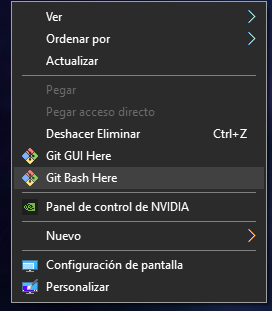
### Linux

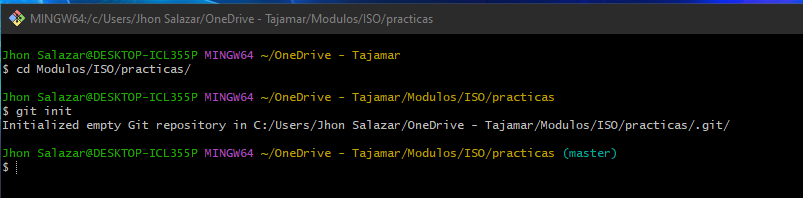
En el terminal iremos a un directorio en el que queramos convertir los archivos que tengamos de proyectos o códigos en repositorios, para ello tendremos que dirigirnos a la ubicación directa por medio de la ruta de comandos y cuando estenmos en la deseado aplicaremos el comando: “git init”. Que nos inicializara el directorio como un repositorio para la subida de archivos a GitHub.

### Windows

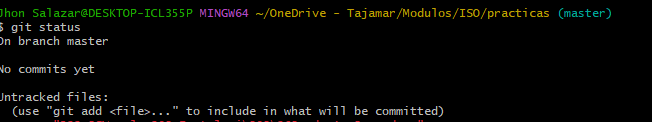
En Windows tendremos que descargarnos de forma externa la herramienta de Git, su software, para integrarlo en nuestro propio sistema operativo, para ello podremos encontrar la última versión en la siguiente página: <https://git-scm.com/download/win>

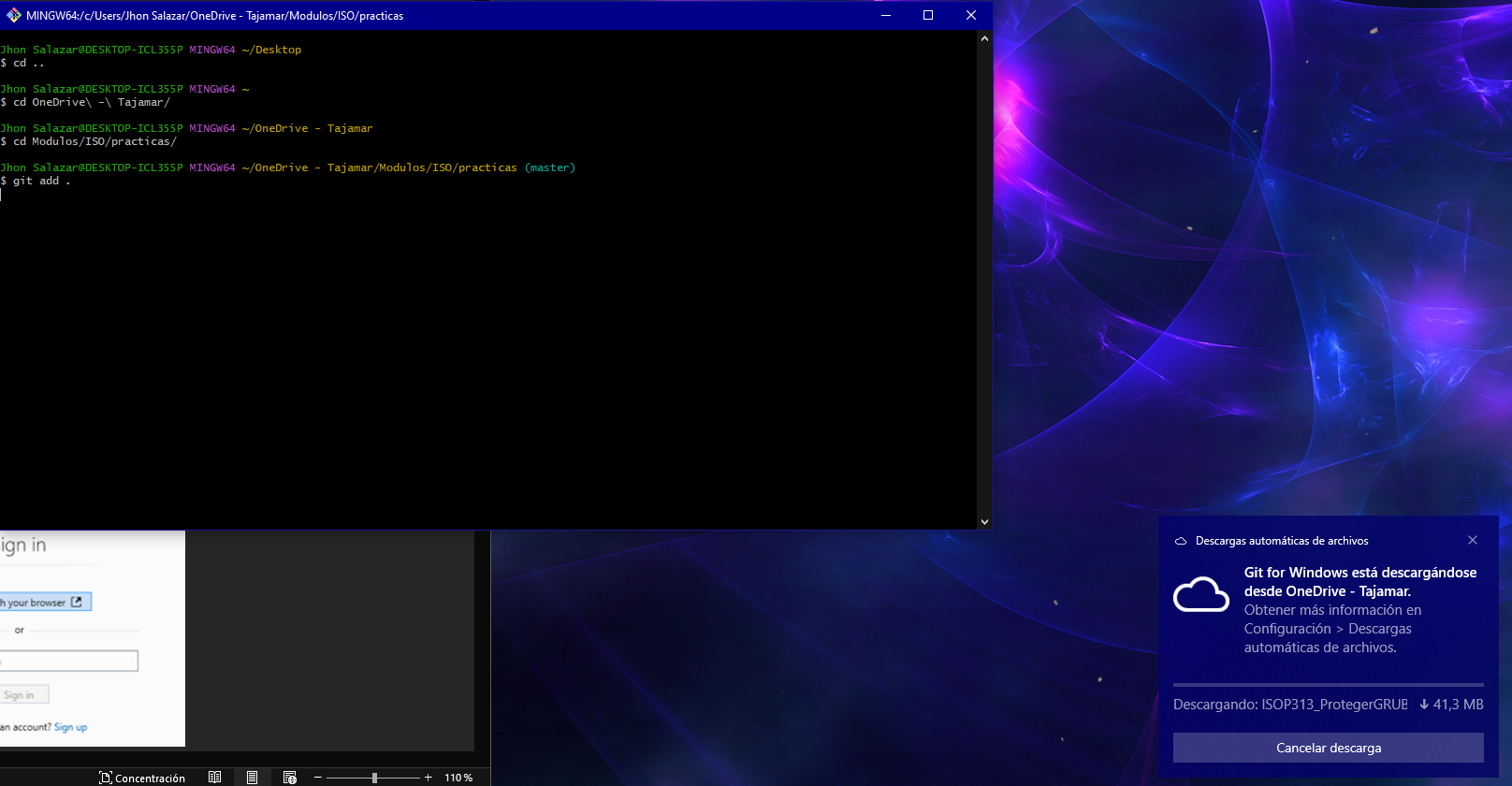
Seguido de esto, habremos descargado el software de Git en Windows y podremos instalarlo en nuestro sistema para poder utilizar la terminal de Git e inicializar las carpetas que tengamos los proyectos que queramos subir a los repositorios.

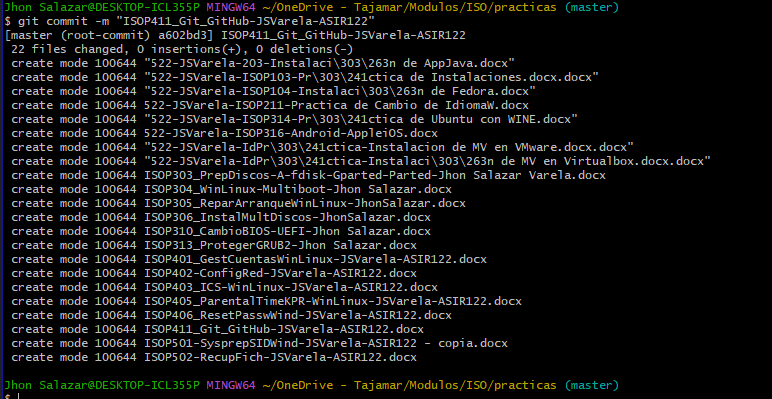


De la misma manera que en Linux, nos iremos al directorio con el que tengamos la intención de crear nuestros archivos y la inicializaremos como un repositorio para GitHub, utilizando el comando de Git “Git init”. (Como se muestra en la siguiente captura, he inicializado el directorio en el que tengo guardadas todas las practicas del curso).

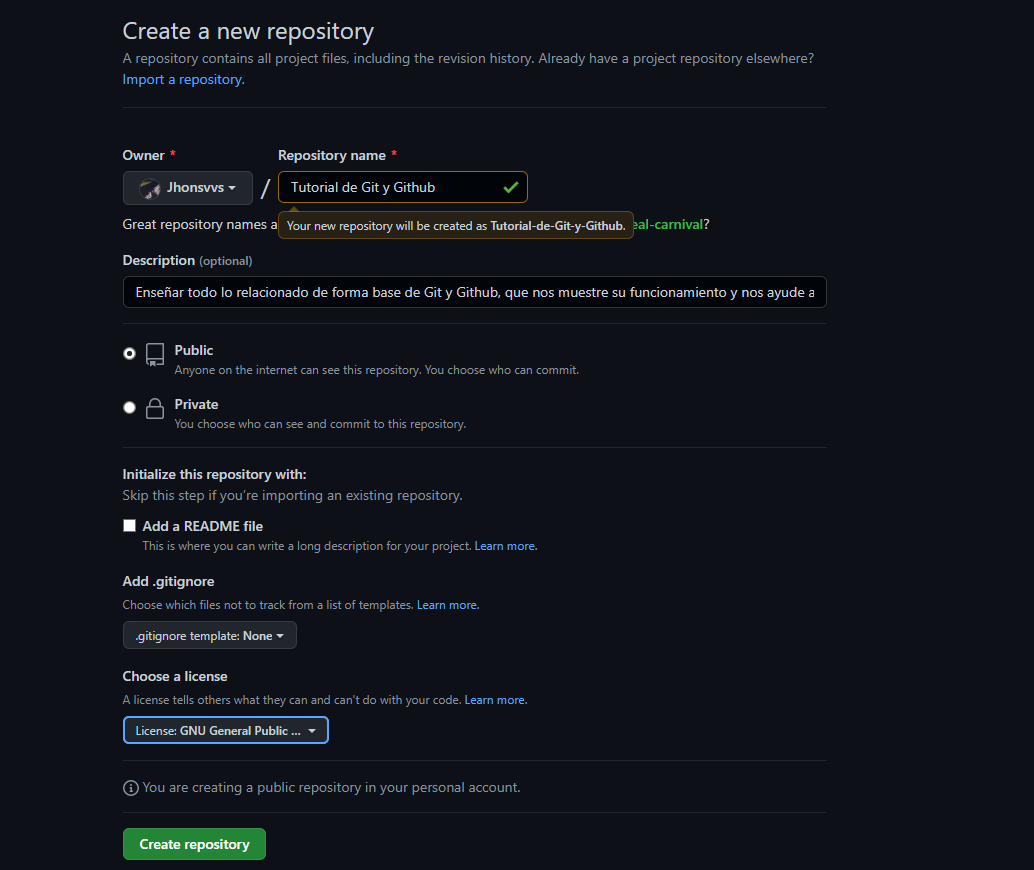
Haremos un “git status” para comprobar el estado del directorio, que nos dirá que no existe ningún comit aun, por lo que nos pedirá que utilicemos el comando “git add” para añadir un archivo que subir al repositorio.



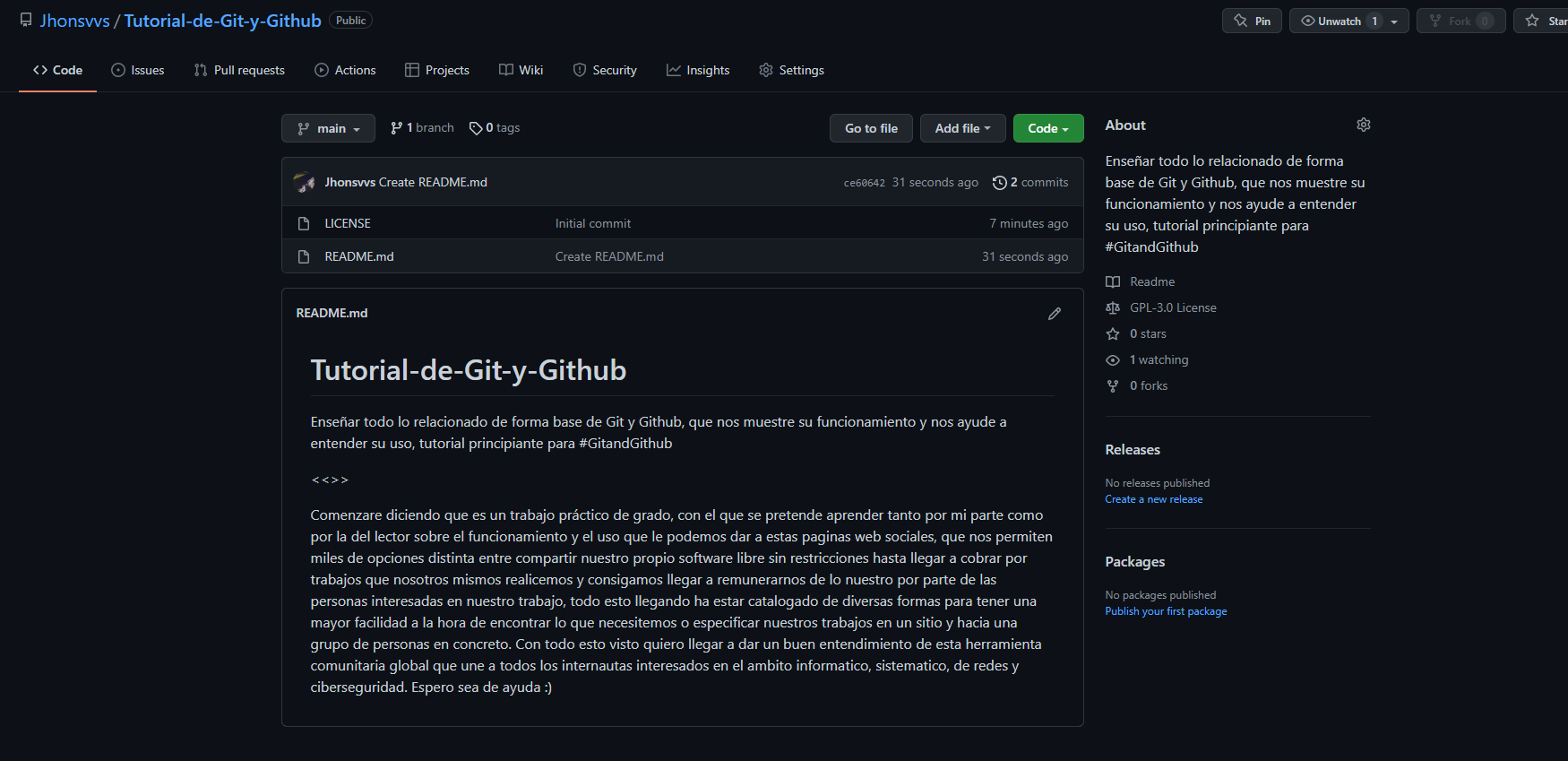
Al ejecutar “Git add .” empezaremos ha añadir todos los ficheros que tengamos creados en el directorio para inicializarlos hacia un repositorio de GitHub.

Con “git commit -m “nombre” crearemos el repositorio en el que incluiremos todos los ficheros mencionados con “git add”, por lo que en mi caso serán todos.

## Por Plataforma

En nuestra cuenta personal desde la pagina web, podremos crear repositorios de la siguiente manera, nos iremos a nuestro perfil y podremos seleccionar en la opción “New”, el apartado de “Repository” nos saldrá una pestaña consiguiente:

Aquí podremos indicar el nombre del repositorio, si será publico o privado (al ser una cuenta gratuita solo podemos hacerlo publico” y decidir si integrar archivos adicionales de información “Readme”, además de una serie de reglas regidas por Github como una licencia que catalogara nuestro repositorio en una búsqueda según las preferencias y a lo que este enfocado. (En mi caso tengo licencia de GNU).

aquí podemos la pagina inicial del repositorio en el que veremos una descripción que hayamos puesto sobre lo que ira nuestro repositorio, así como los ficheros de adicionales creados junto con este.

Podremos subir nuestros archivos desde nuestra carpeta directamente para el repositorio creado a través de la siguiente opción:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

# Consideraciones finales y metacognición