

## USO DEL GITHUB

**GIT** es el software que rastrea. El sistema de control de versiones. La herramienta que utilizaremos en la terminal.

**GitHub** es la plataforma de "hosting" de los proyectos. Una comunidad llena de personas que desarrollan y comparten, usando GIT.

1.- Vamos a utilizar GitHub, necesitarás registrarte con una cuenta en [GitHub](https://github.com).

2.- Descarga el GitHub DeskTop

### 3.- PASOS PARA USAR GitHub DeskTop

#### Crear un documento

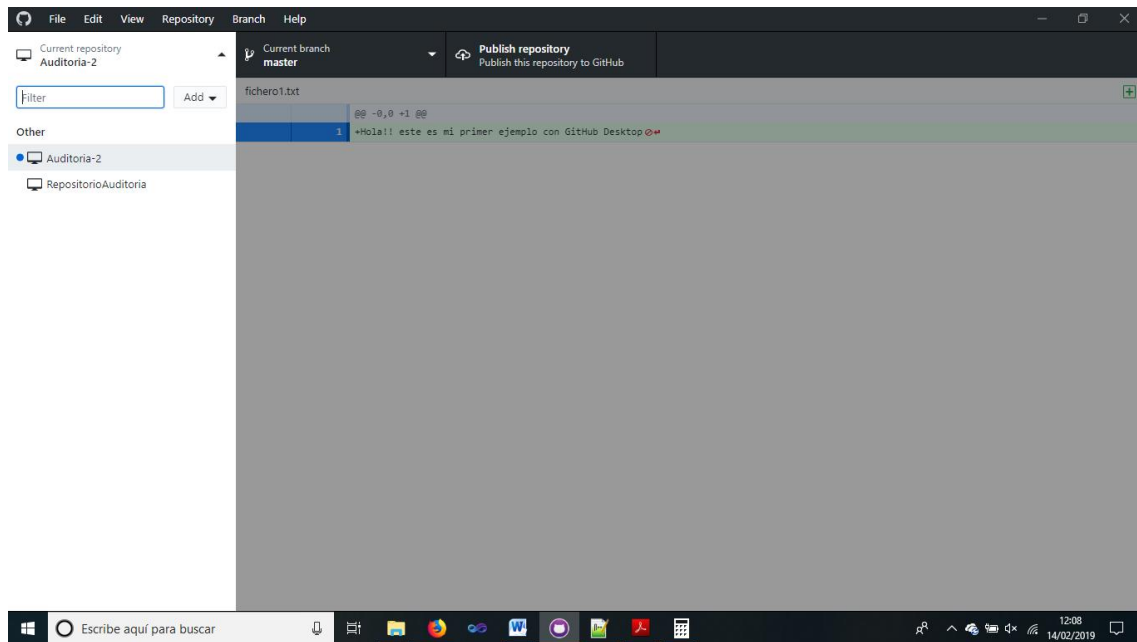
Podemos empezar creando un documento muy sencillo.

```
;Hola mundo!
```

#### Crear un repositorio y añadir un documento

Hay varias formas de *añadir* un archivo para que GitHub Desktop lo controle.

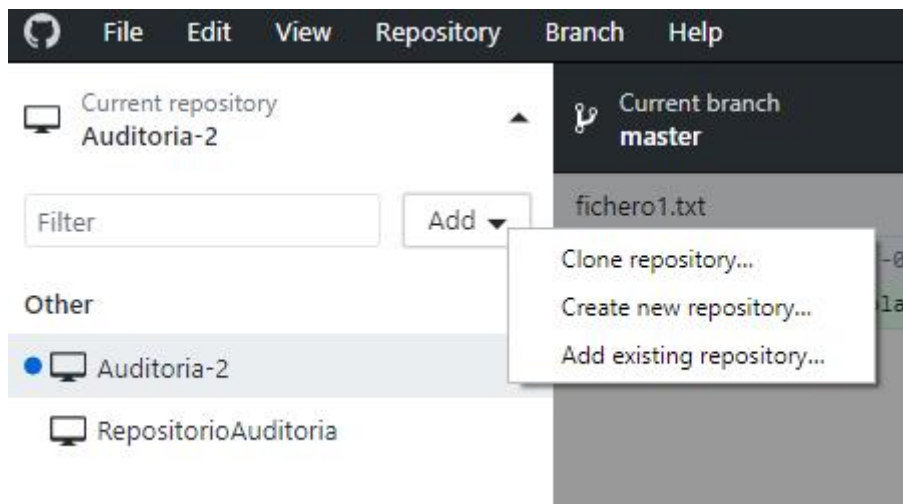
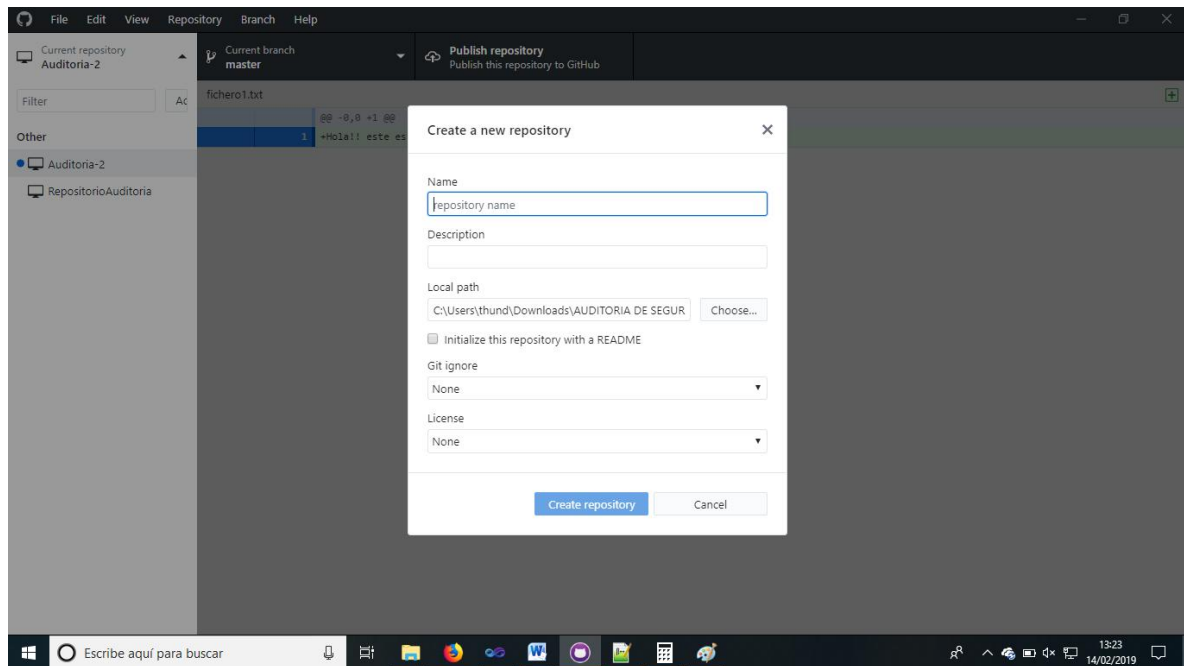
El primer paso es crear un repositorio, para ello situarse en el panel de la izquierda y pulsar el botón Add:



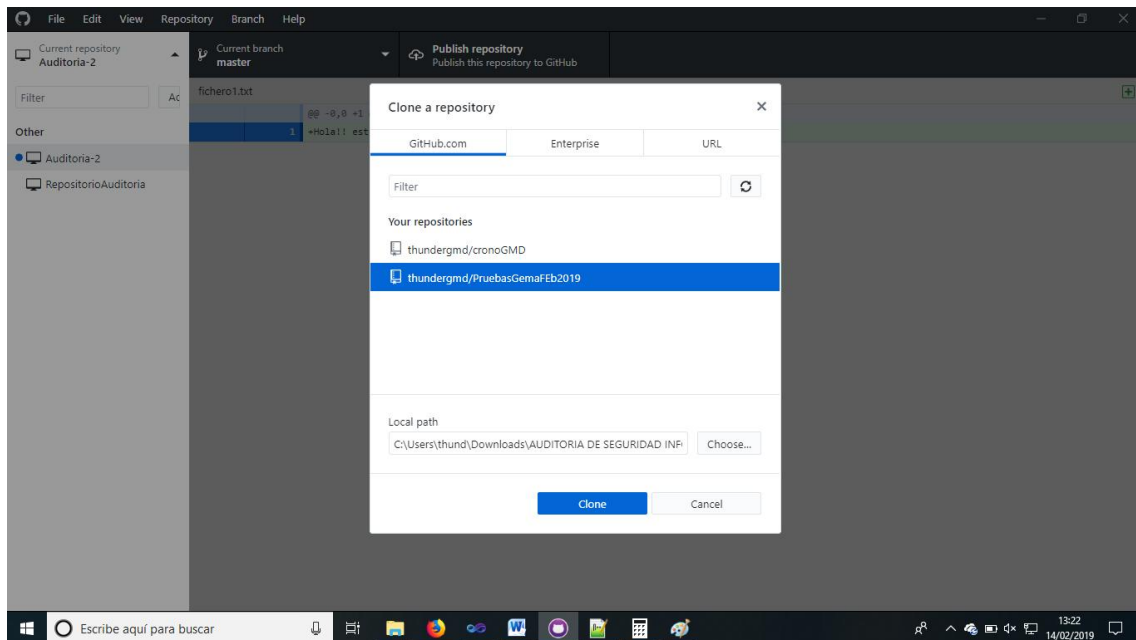
Tenemos tres opciones diferentes para añadir un repositorio para nuestros ficheros:

- Clone repositorio: duplicar un repositorio existente.
- Create new repository: crear un nuevo repositorio
- Add existing repository: añadir repositorio existente

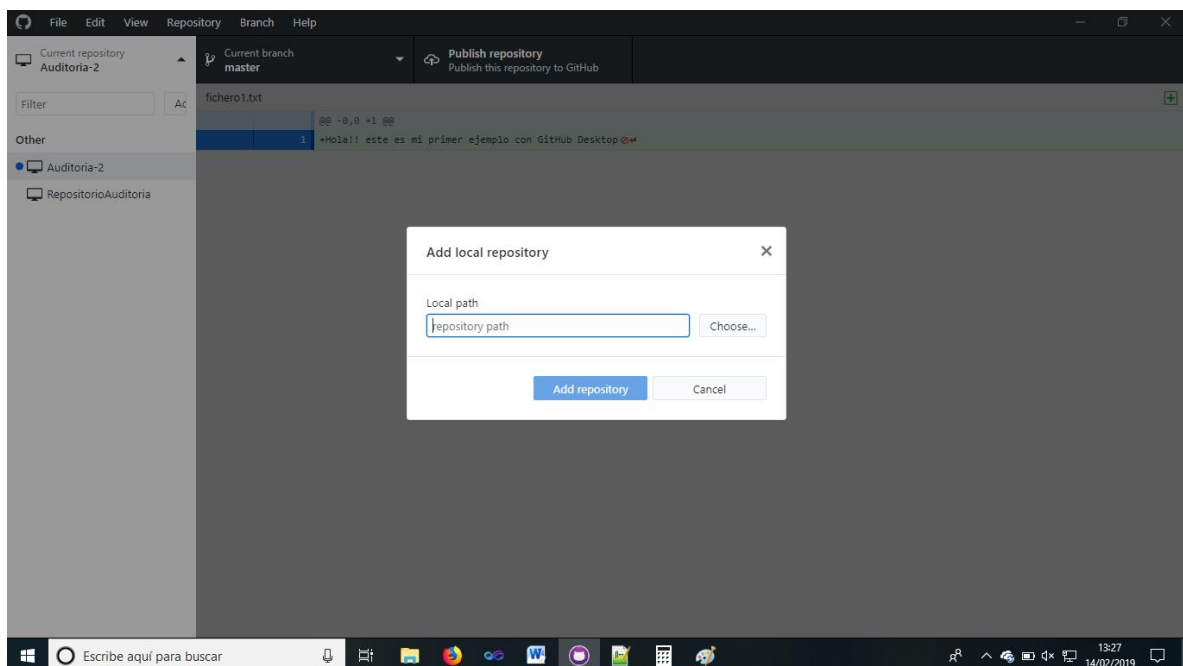
**Create new repository:** esta opción crea un nuevo repositorio indicando el nombre, la descripción, el path local donde residirán los ficheros del mismo. Existe la posibilidad de elegir una de las licencias de la lista (no es obligatorio) y además podremos asignar aquellos ficheros que queremos que evite el Git en “Git Ignore”.



**Clone repository:** al pulsar esta opción se mostrará la siguiente interface, que me permitirá duplicar un repositorio que tenga en local. Para ello debo seleccionarlo de la lista de repositorios disponibles y pulsar el botón Clone:



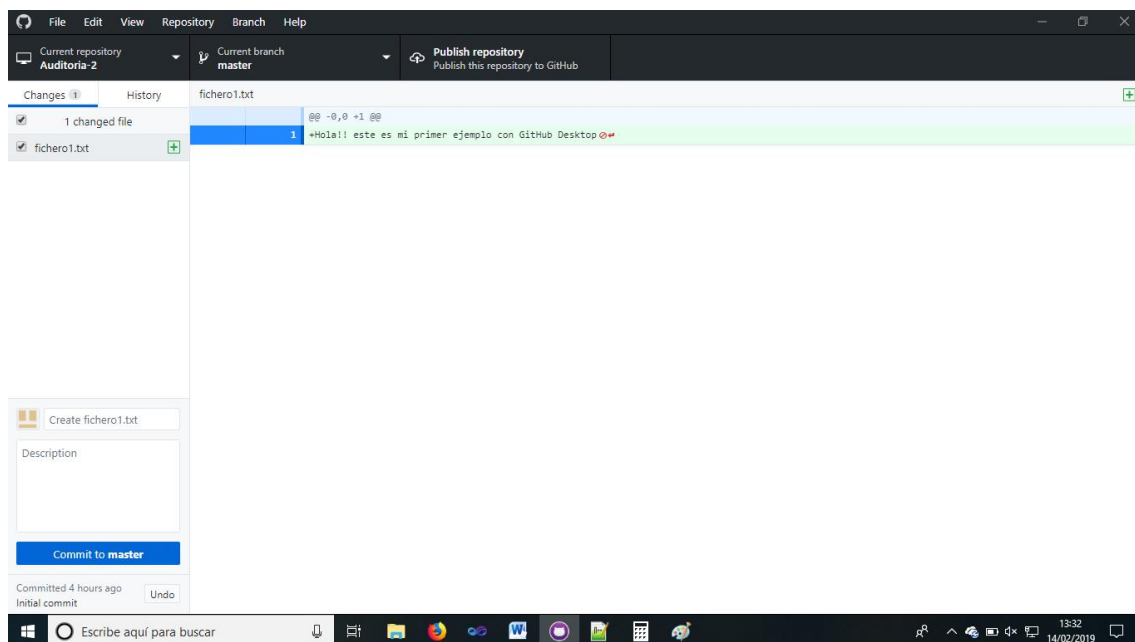
**Add existing repository:** debemos indicar la ruta local del repositorio a añadir.



Crearemos 3 repositorios diferentes para probar las opciones anteriores. La ruta local de cada repositorio deberá contener los ficheros sobre los que vamos a aplicar el control de versiones.

## Anotar cambios

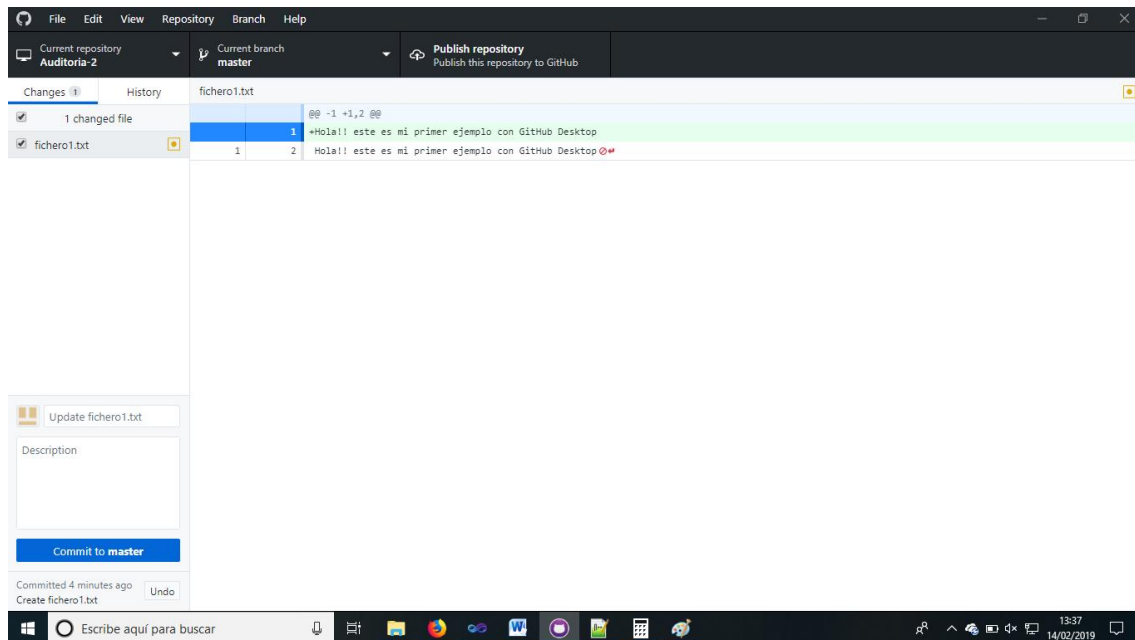
Al *anotar* ('commit') un cambio, comunicas a Git que quieres registrar las modificaciones realizadas. Aunque *anotar* puede parecer similar a guardar un archivo, el objetivo es distinto. A menudo guardamos diferentes versiones de un documento; ahora bien, guardar un documento, en realidad, significa que puedes cerrar el archivo y volver a él más tarde y que su estado será el mismo, es decir, no se habrán producido pérdidas. *Anotar*, en cambio, implica tomar una instantánea de un archivo en un momento determinado y documentar información sobre los cambios realizados.



## Primera anotación

Para anotar un cambio debes dar un resumen de los cambios y, de manera opcional, incluir una descripción. Es importante que pienses con cuidado cuándo debes anotar los cambios. El control de versiones solo es útil si anotas los cambios de manera eficiente. A veces tendemos a anotar los cambios solo cuando hemos terminado de trabajar sobre un documento. Sin embargo, esto no refleja los cambios importantes realizados durante todo el proceso.

Debemos crear varios ficheros sencillos de texto, una vez editados y guardados, veremos la siguiente ventana sobre el programa:



A la izquierda hay dos solapas:

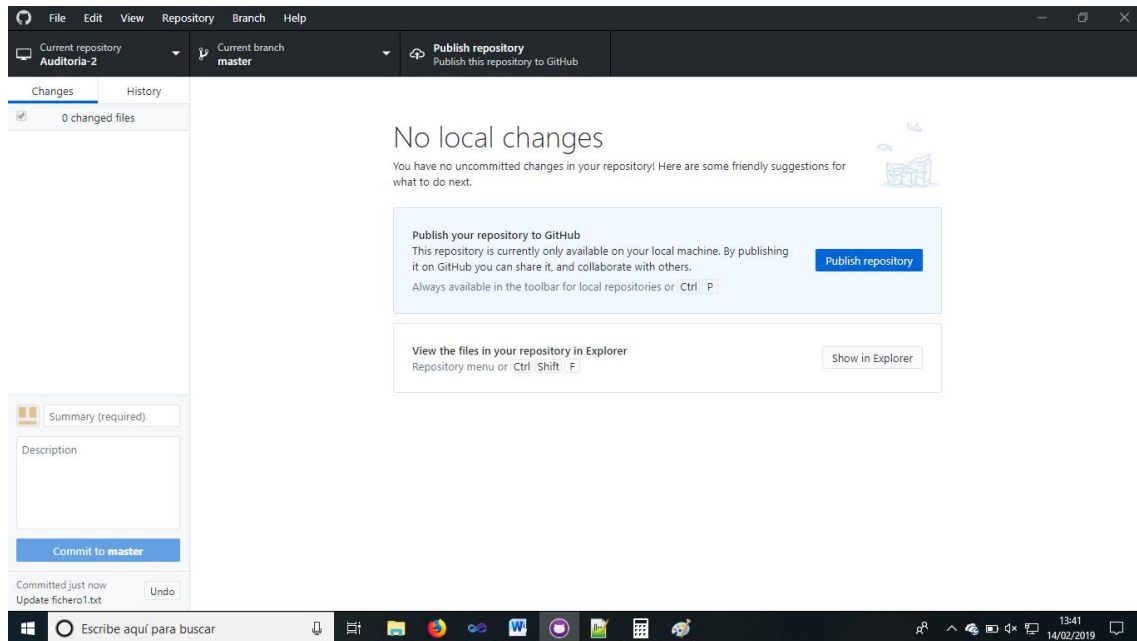
**1.-Changes:** indica los cambios que debes guardar en el repositorio; se creará una instantánea de dicho momento con los cambios abordados en el documento. En la parte derecha de la ventana aparece con una flecha roja aquellas líneas del fichero que han sido modificadas en el mismo:



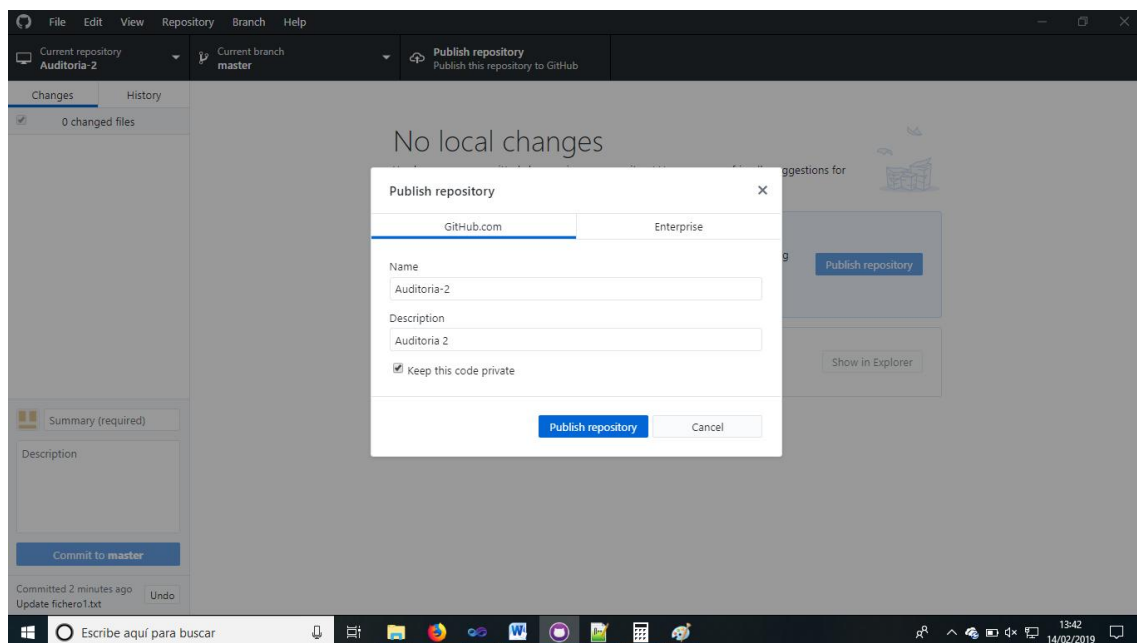
**2.-History:** historial de los cambios producidos en el fichero o ficheros.

Para realizar esta anotación debemos pulsar el botón de **Commit to master**. Cuando anotes el cambio verás que aparece el mensaje ‘anotar al master’. Esto quiere decir que te refieres a la rama ‘master’. En un repositorio Git es posible tener varias ramas. Estas ramas son, en esencia, lugares distintos en los que puedes trabajar. A menudo se utilizan para probar nuevas ideas o trabajar en un aspecto concreto. En principio, no es necesario utilizar crear ramas en GitHub pero quizás quieras aprender de cara al futuro, sobre todo si deseas colaborar con otras personas en un mismo proyecto.

Al pulsar se guardan los cambios en nueva versión y aparece la ventana:

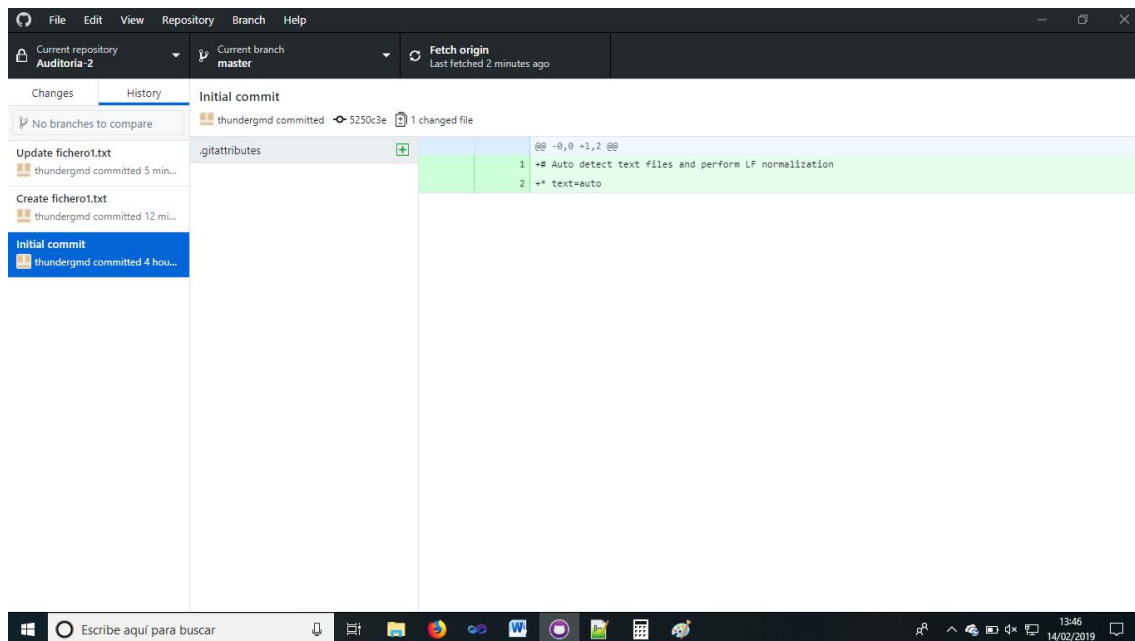


Ya no existen cambios en el fichero elegido pero puedes publicar los cambios en tu repositorio de GitHub, es decir, en la nube, ya que hasta ese momento están en local:



Al dar a publicar repositorio tenemos la opción de hacerlo en GitHub.com o en la versión profesional Enterprise (nosotros usaremos GitHub).

**History:** historial de los cambios producidos en el fichero o ficheros:



En la parte izquierda vemos que aparecen las distintas versiones del documento; pulso cualquiera de ellas y a la derecha me informa de los cambios producidos en el fichero.