

2019

# Tutorial GIT



**git**

Jose Camacho

MIM Patchwork Service

02/12/2019

## Contenido

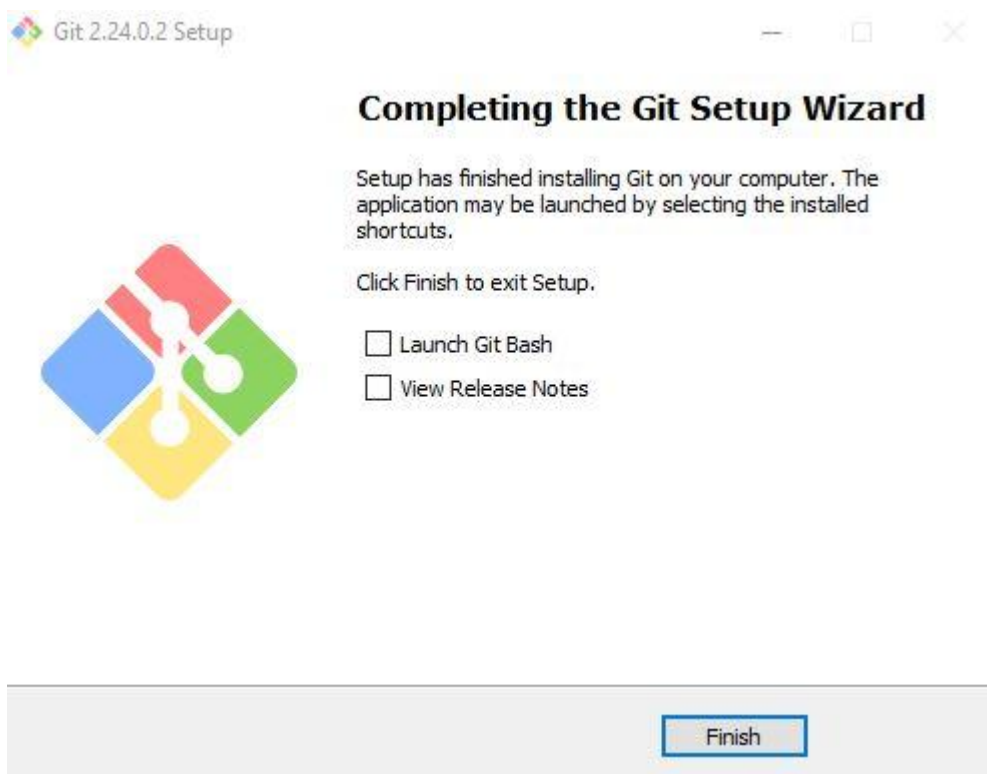
1. Instalación de Git.....	3
2. Crear proyecto.....	6
3. Clone.....	7
4. Commit.....	11
5. Pull.....	13
6. Push.....	14
7. Revertir cambios .....	15

## 1. Instalación de Git

Lo primero que tenemos que hacer es descargar e instalar Git en nuestro equipo, para ello accedemos a [www.git-scm.com/downloads](https://www.git-scm.com/downloads) y descargamos la versión que necesitamos, en nuestro caso será para Windows.



Para instalarlo simplemente iremos haciendo Next a través del menú de instalación, sin cambiar ninguna opción de las que vienen por defecto.



Lo siguiente será instalar el cliente para Git, hemos elegido SmartGit por su facilidad de uso y su integración con repositorios privados. Para descargarlo accedemos a [www.syntevo.com/smartgit/download/](https://www.syntevo.com/smartgit/download/) y descargamos la versión para Windows.

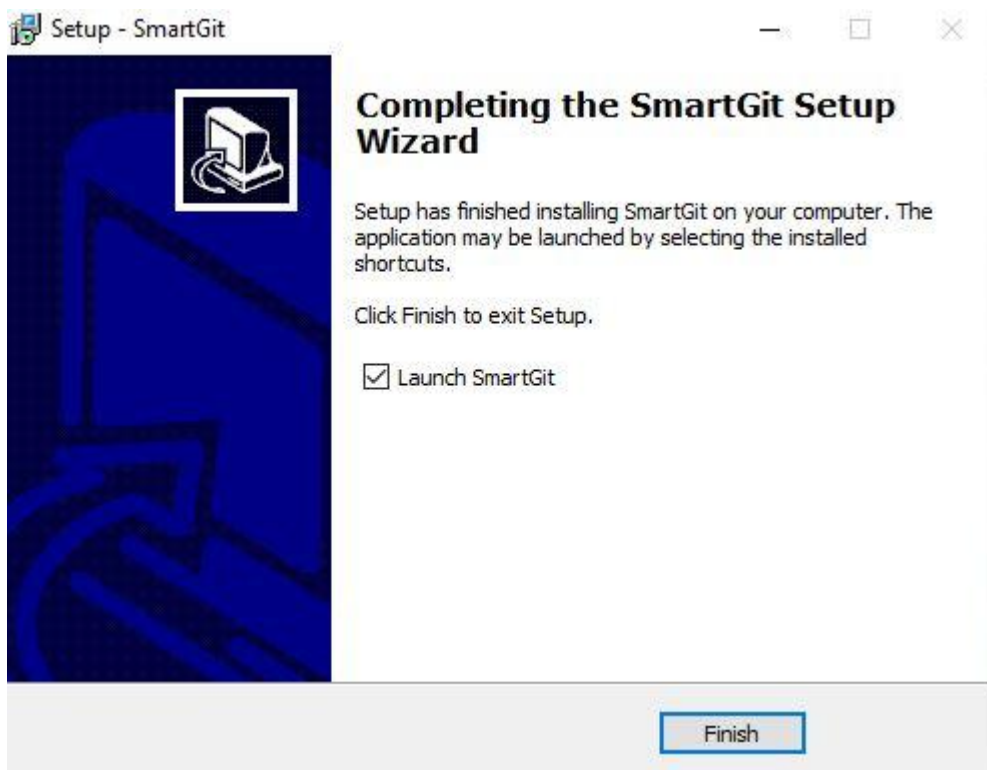
## — Download SmartGit 19.1

Version 19.1.4 ([Change Log](#))

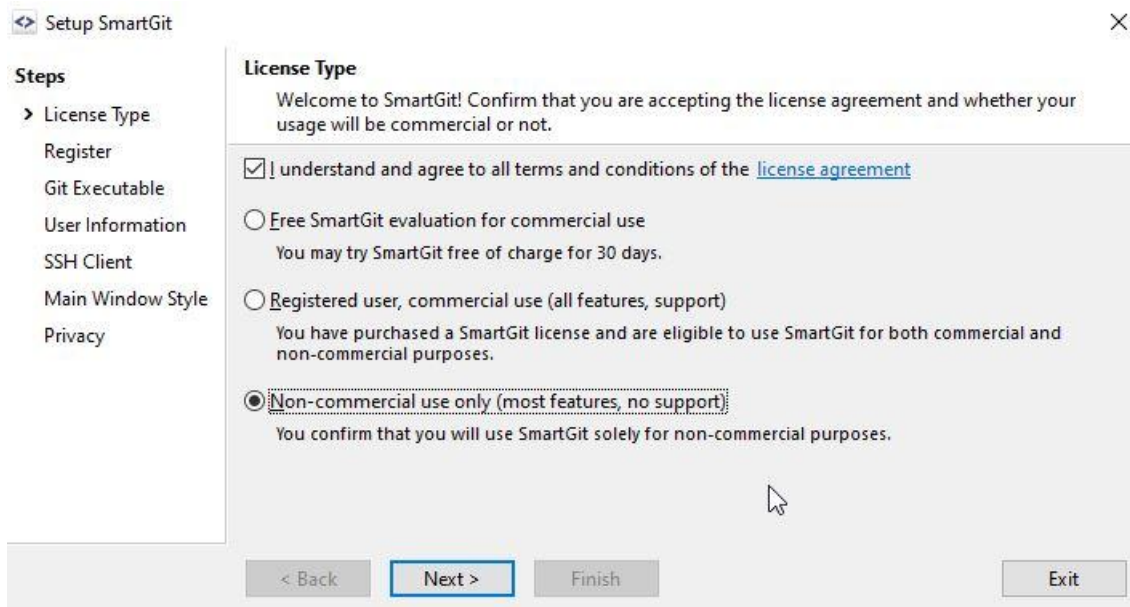
[Preview Version](#) [Archive \(for older versions\)](#)

<b>Download for Windows</b> Windows 7+ (64-bit), Git, Installer ⓘ <b>Portable Bundle</b> Windows 7+ (64-bit), Git, Archive (7z) ⓘ	<b>Download for macOS</b> macOS 10.11 - 10.14, Git, Archive (dmg) ⓘ	<b>Download for Linux</b> Linux (64-bit), Archive (tar.gz) ⓘ <b>Debian Bundle</b> Linux (64-bit), Archive (deb) ⓘ
--	--	--

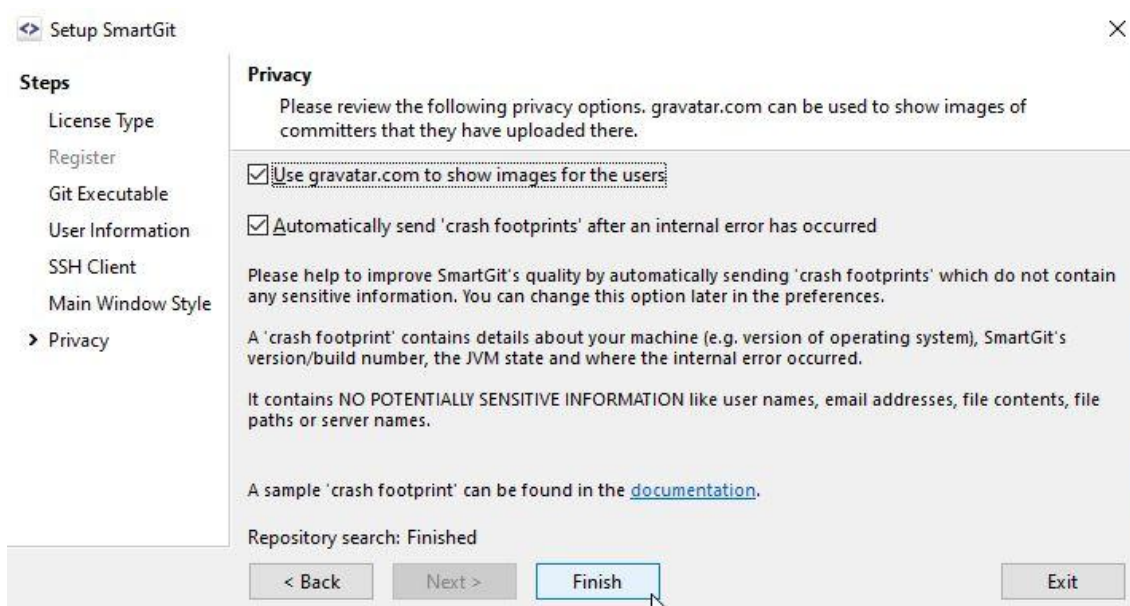
Esto nos descargará un archivo comprimido con un ejecutable dentro. Comenzamos la instalación, en el paso que nos indica si queremos crear un acceso directo en el escritorio, marcamos esa opción. Seguimos los pasos con Next hasta completar la instalación.



Al completarse, nos aparecerá un recuadro para configurar el cliente. Marcamos la opción "I understand and agree to all terms and conditions of the license agreement" y marcamos la opción "Non-comercial use only".

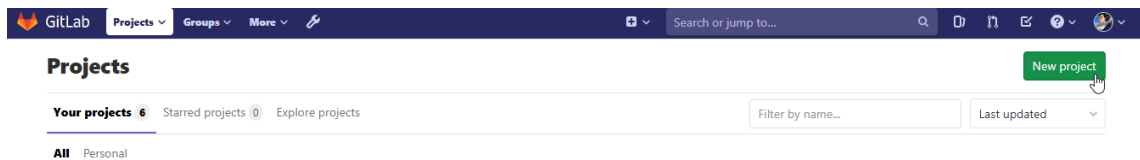


Hacemos click en Next, marcamos las dos casillas y continuamos con OK. Las siguientes opciones podemos dejarlas por defecto y avanzar con Next hasta el Finish. Y con esto ya tendremos todo instalado.

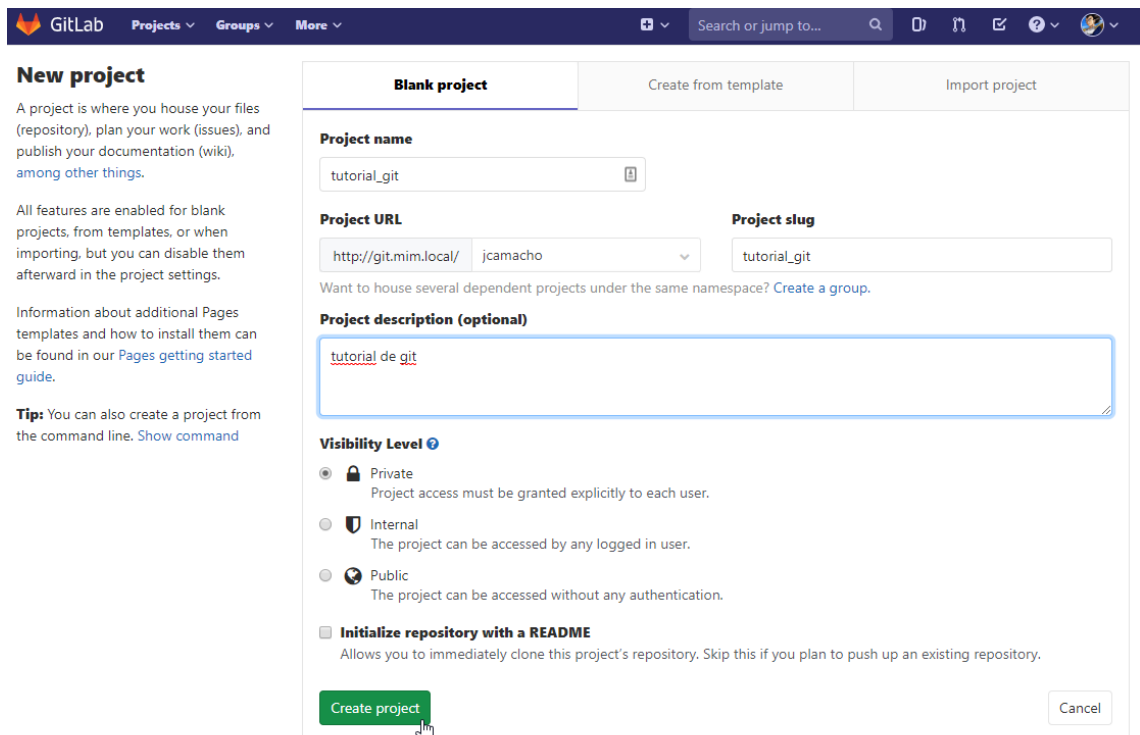


## 2. Crear proyecto

Para crear un proyecto lo haremos accediendo a la url de nuestro repositorio <http://git.mim.local/> aquí debemos hacer login con nuestro usuario y contraseña. Después haremos click sobre New project y nos abrirá la página para crear un nuevo repositorio.



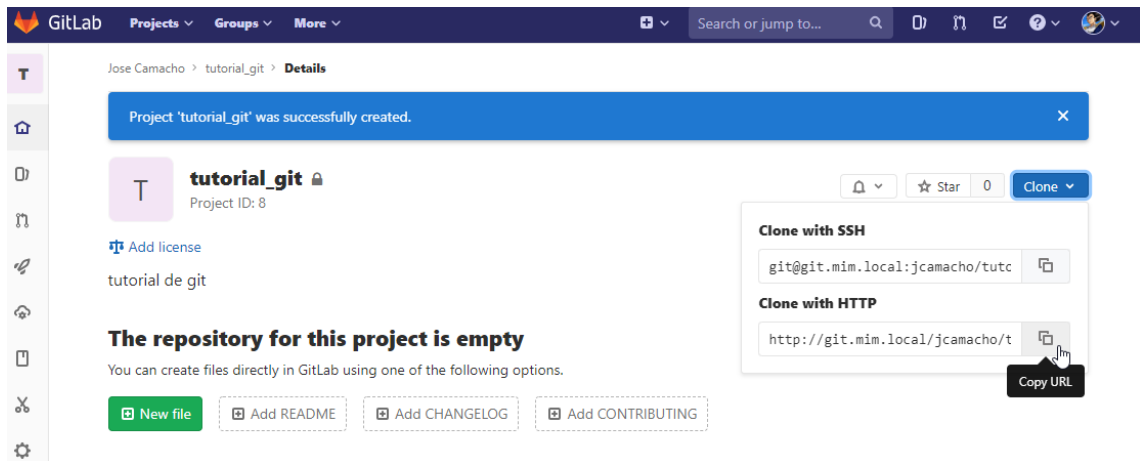
Rellenamos los datos necesarios y clickamos sobre Create project para generar un repositorio vacío.



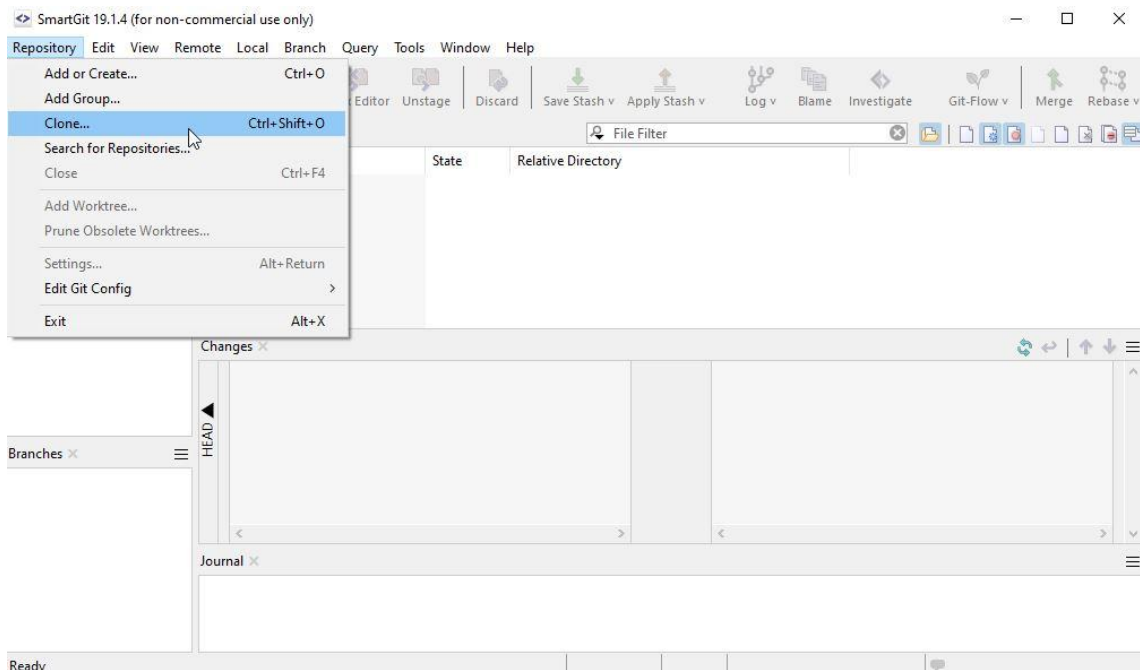
### 3. Clone

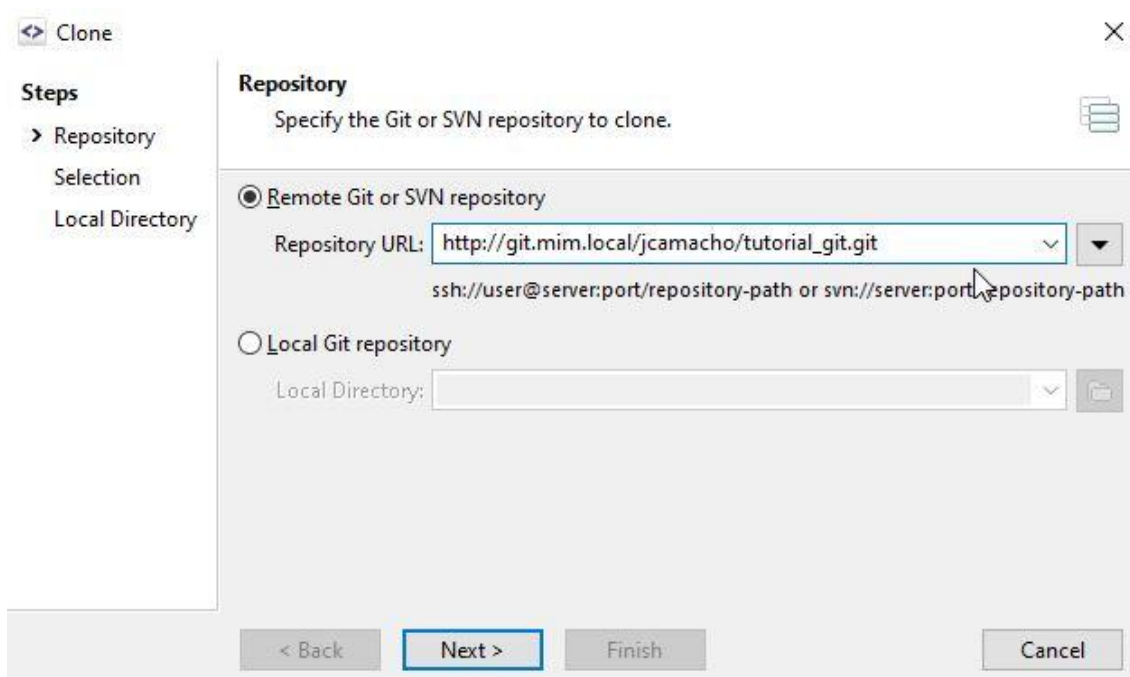
La acción clone es la que se suele utilizar generalmente cuando alguien quiere reengancharse a nuestro código del repositorio o cuando queremos compartirlo. Para ello necesitamos una dirección HTTP.

En el ejemplo, vamos a hacerlo con el nuevo repositorio que hemos creado en este tutorial en el punto 2. Nos dirigimos al proyecto y hacemos click en Clone, y a continuación copiamos la dirección HTTP.

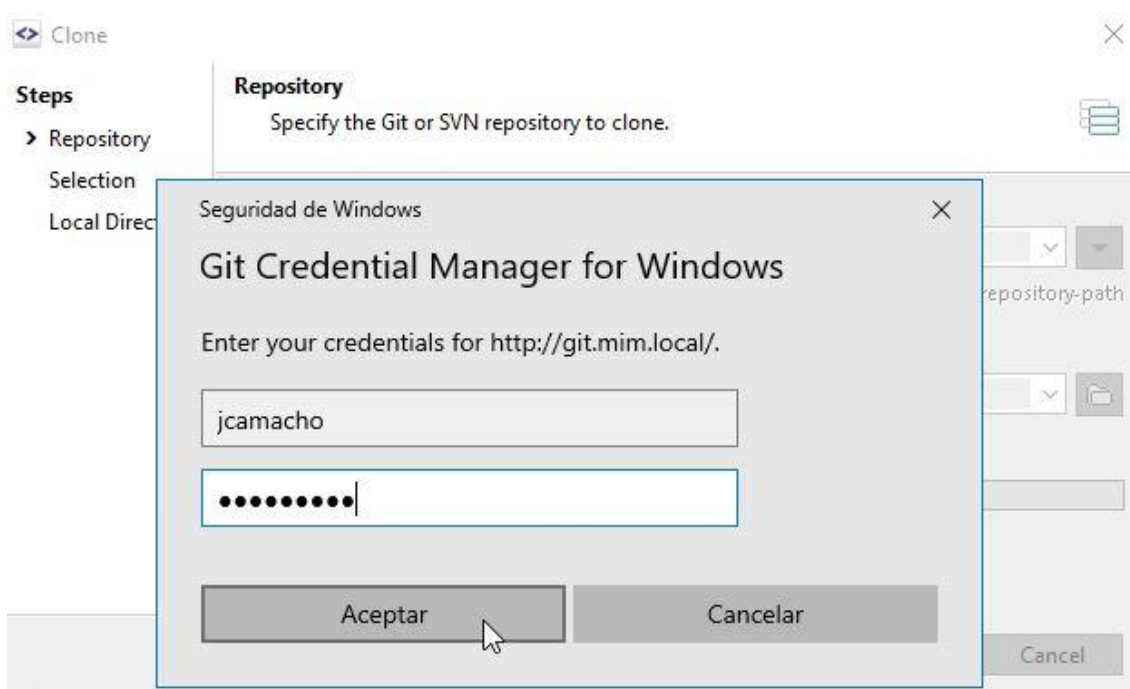


Ahora es donde empezaremos a utilizar el cliente SmartGit que hemos instalado. Le daremos sobre Repository -> Clone... y pegamos la dirección que acabamos de copiar.



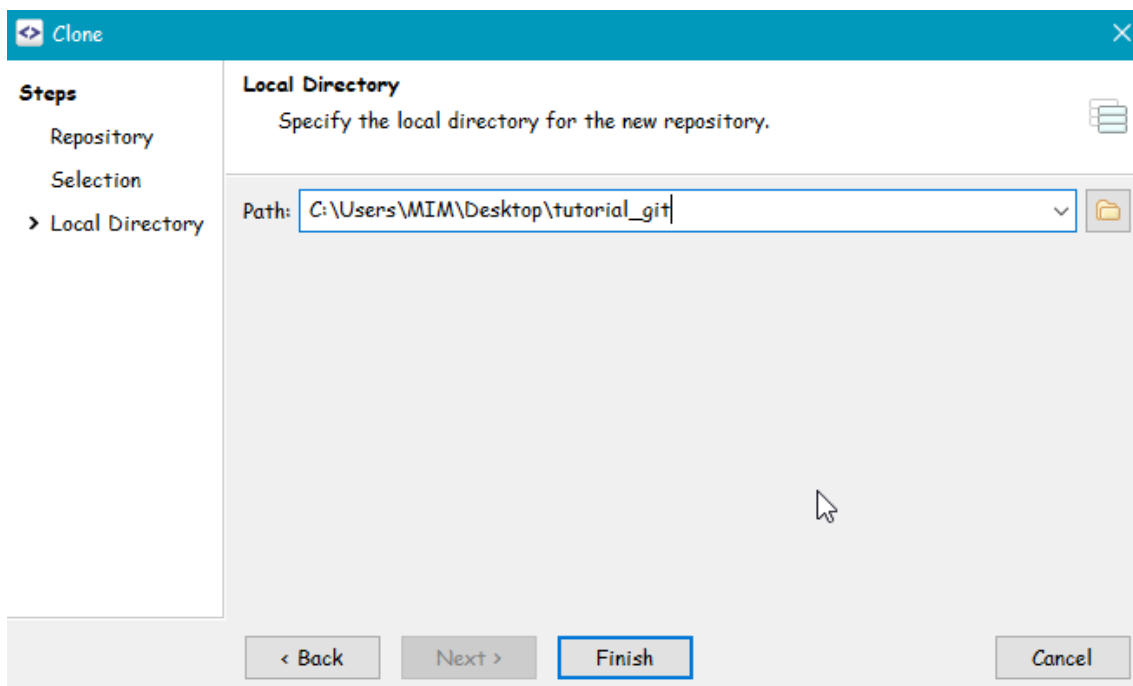
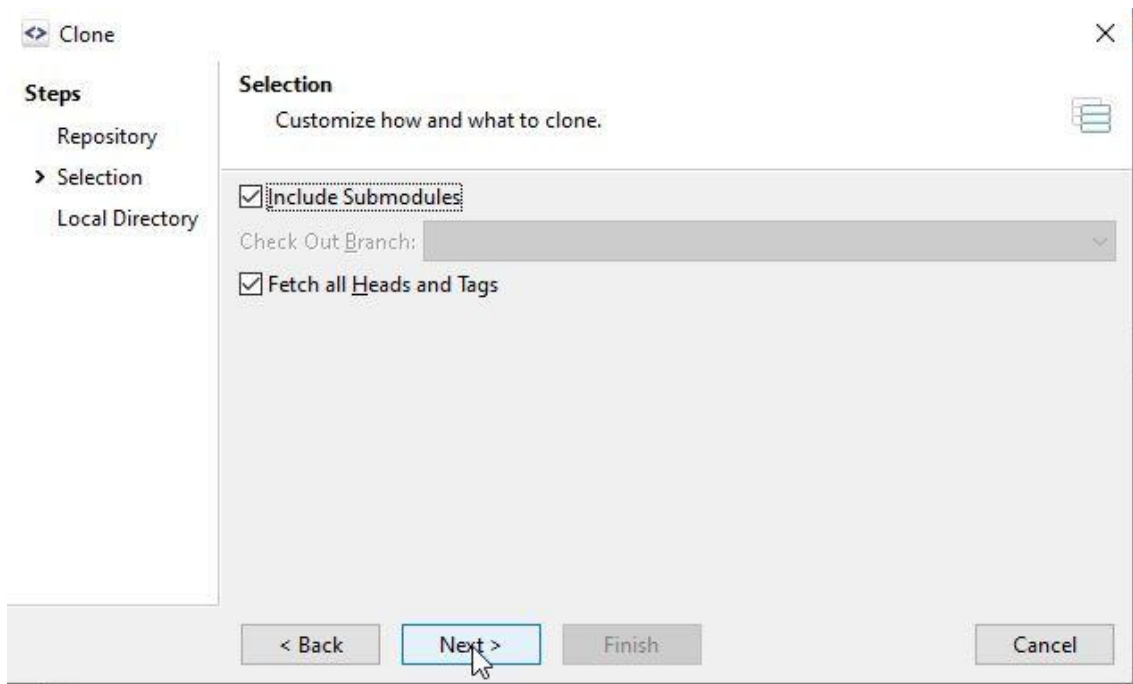


Al ser la primera vez que entramos con el cliente Git, nos pide autenticación. Hacemos login con el usuario y contraseña que hemos utilizado para entrar en la web del repositorio.

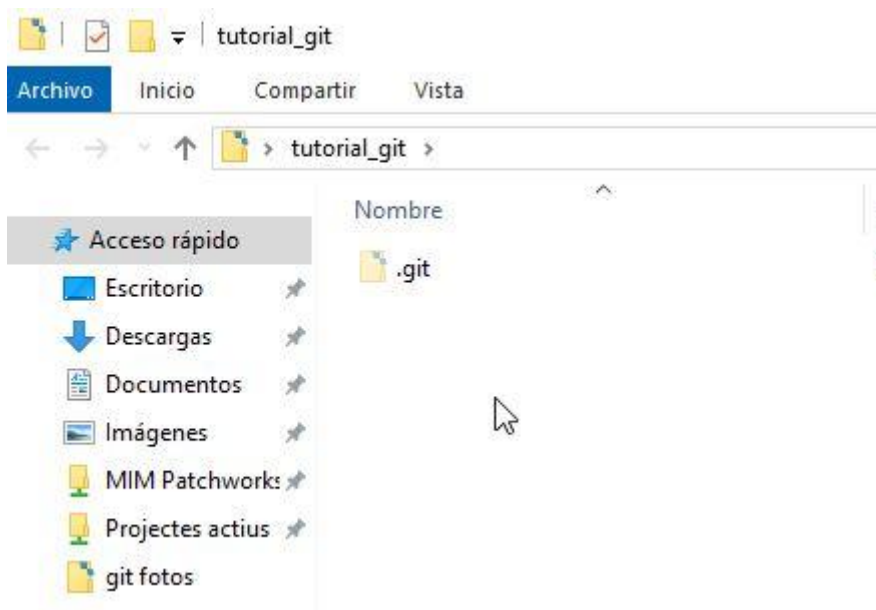
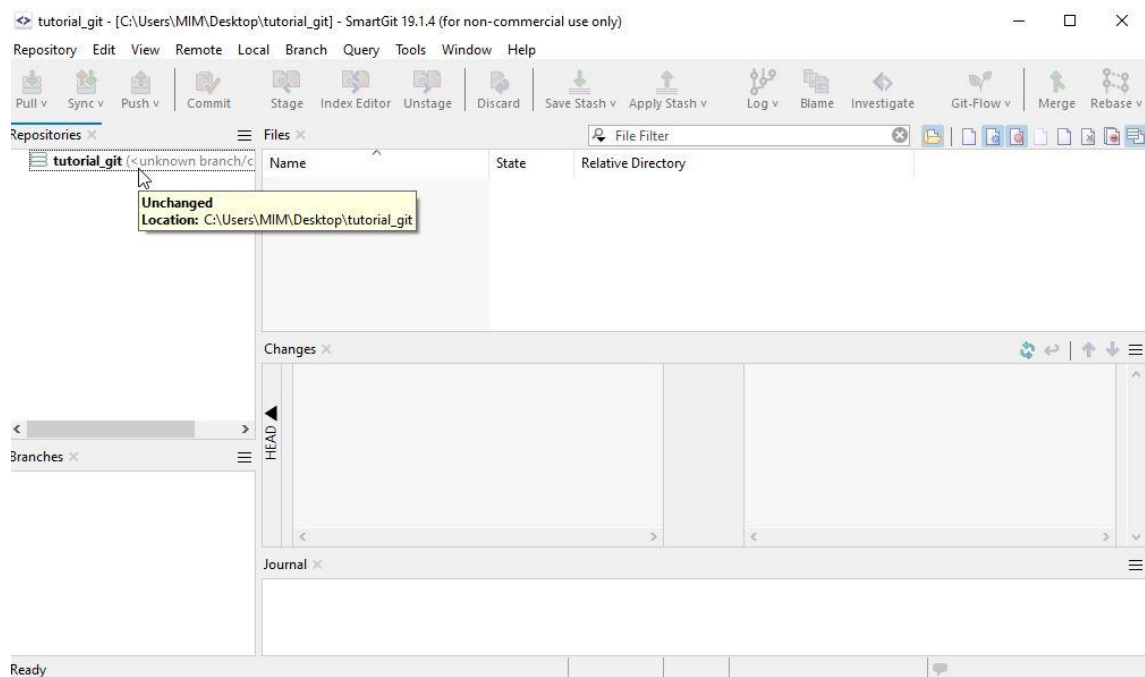


Hacemos click en Next y a continuación indicamos la carpeta nuestra en local donde queremos que nos haga un clon del proyecto. La seleccionamos y hacemos click en Finish.





Ahora ya nos aparece el repositorio en el SmartGit y en nuestra carpeta local se habrá clonado lo que contenga el repositorio remoto en ese momento. También podremos apreciar que hay una carpeta oculta con el nombre de .git que contiene los archivos de configuración del repositorio.



## 4. Commit

La acción commit sirve para guardar los cambios que hayamos realizado en el proyecto. Estos cambios se guardan en local hasta que decidamos subirlos al repositorio mediante la acción push.

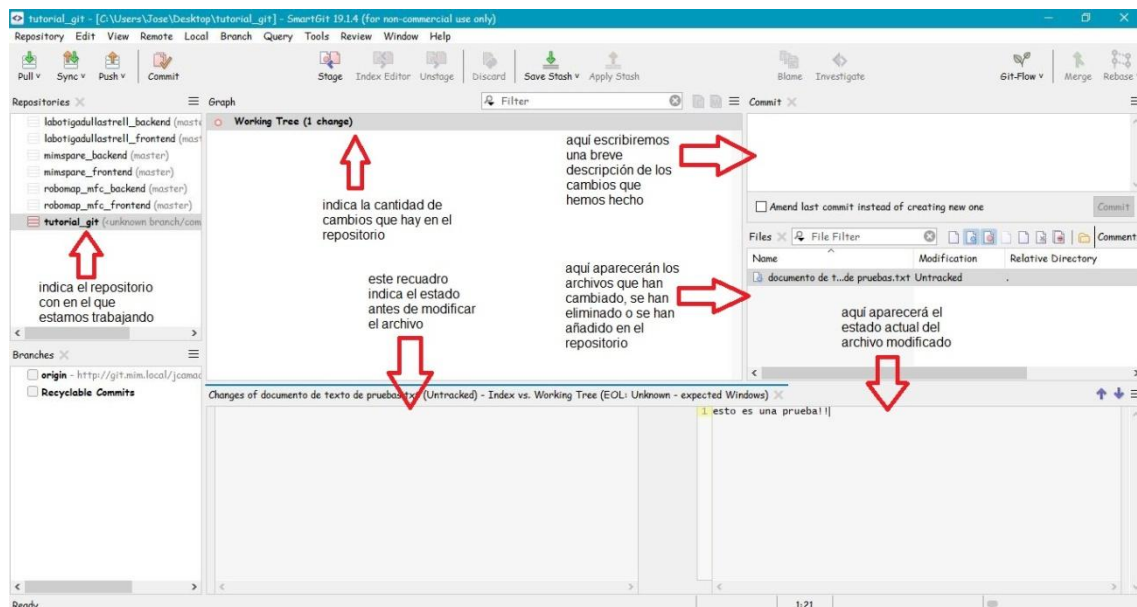
Se pueden almacenar tantos commit como se deseen antes de realizar el push, después se subirán todos en orden de fecha.

Cada uno de estos commits los diferenciaremos con un título o descripción en el que trataremos de detallar el por qué de este guardado.

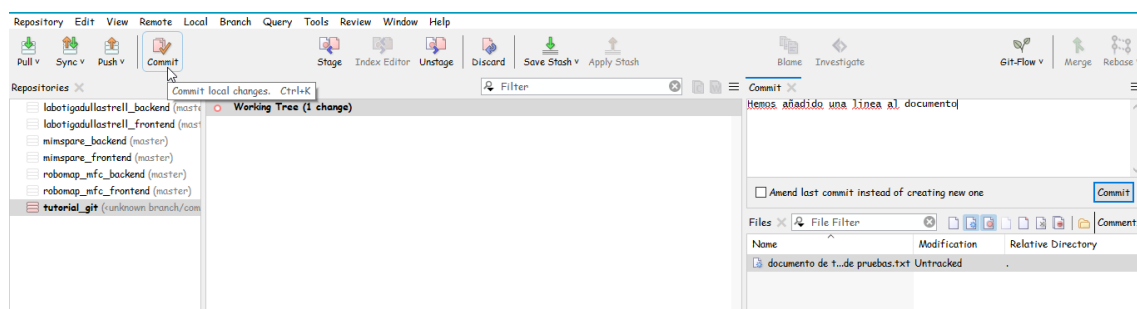
De esta manera, si tenemos un error y queremos volver a la versión anterior a este cambio, lo tendremos identificado.

En el ejemplo vamos a modificar la información de un archivo de un archivo de texto que creamos en la carpeta del proyecto.

Aquí ya podremos observar la siguiente información:



Lo siguiente será hacer un comentario sobre el cambio que hemos realizado y hacer click sobre Commit.



Marcamos los archivos que queremos que sean guardados y confirmamos el cambio haciendo click de nuevo sobre Commit.

Commit

Commit local or staged changes

Select the files you want to commit and provide a commit message.

Staged Changes
Local Changes

1 file (+1)

Name	Directory
<input checked="" type="checkbox"/> documento de texto de pruebas.txt	.

Commit Message:

Select

Hemos añadido una linea al documento

☐ Amend last commit instead of creating new one

More Options

Commit

Commit & Push

Cancel

Aparecerá un mensaje de que el commit se ha realizado con éxito, a la derecha aparecerá quien ha realizado el último commit, el momento en el que lo ha hecho y el mensaje que ha escrito al hacerlo.

master
Hemos añadido una linea al documento
JC 09:30

Commit 9ad83223

by Jose Camacho, 2019-12-02 09:30

Hemos añadido una linea al documento

branches master

Files
File Filter

Name	Modification	Relative Directory
documento de t...de pruebas.txt	Added	.

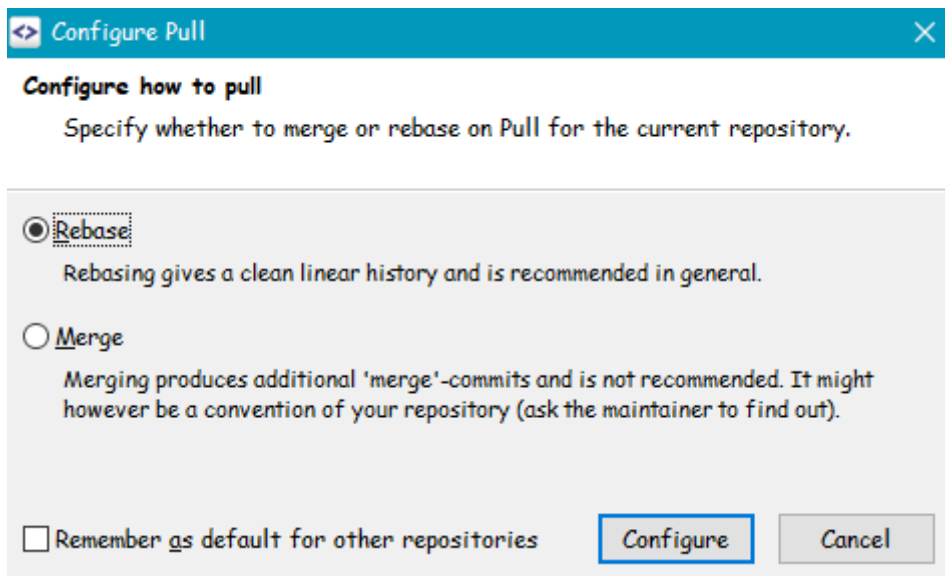
pág. 12

## 5. Pull

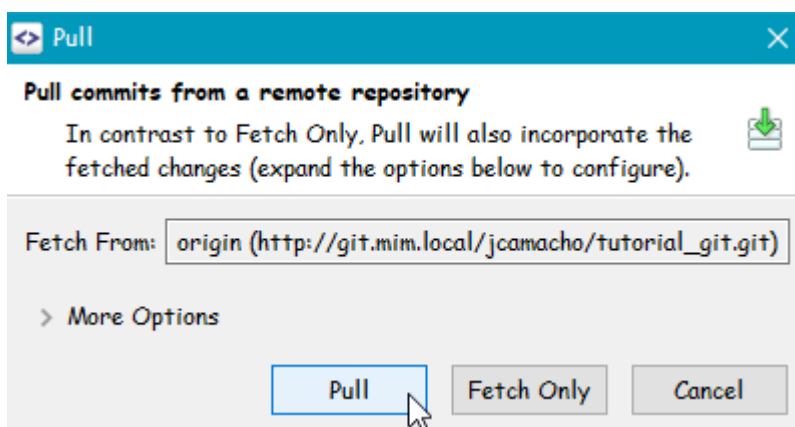
Con la acción pull recogeremos los cambios que se hayan subido al repositorio que aún no tengamos integrados en nuestro código en local.

Es recomendable hacer siempre un pull antes de subir nuestros cambios por si algún compañero ha subido código, poder tenerlo integrado al nuestro antes de subirlo.

Al hacer click sobre Pull, puede que nos aparezca la opción para elegir entre Merge y Rebase. La diferencia es que Rebase te unifica todos los cambios y simplemente aparecerá como que has cogido los cambios existentes, mientras que un Merge no unifica los cambios y los muestra todos. Personalmente preferimos elegir Rebase, ya que queda la cola de cambios mucho más limpia.



A continuació elegimos Pull, y con esto ya habremos cogido los cambios que puedan existir en el repositorio.

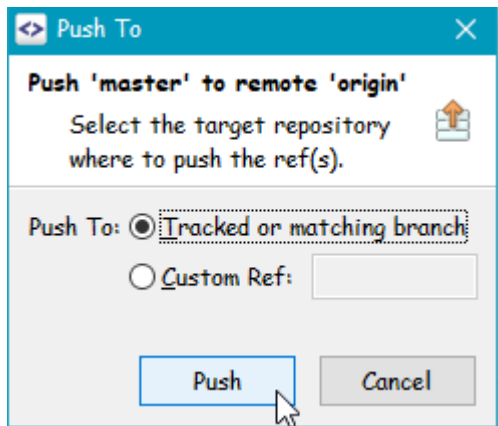


## 6. Push

Con la acción push subiremos los cambios que hayamos guardado en nuestro código en local mediante la acción commit.

De esta forma queda grabado en el historial de commits el usuario que ha realizado el push con la descripción que indicó en el commit.

Para realizarlo, hacemos click en Push, y de nuevo Push en la ventana emergente.




Ahora, si accedemos al repositorio, podemos ver que el commit se ha subido y podemos ver los cambios que se han realizado en él.


---

Jose Camacho > tutorial\_git > **Repository**

master tutorial\_git / +

History Find file Web IDE

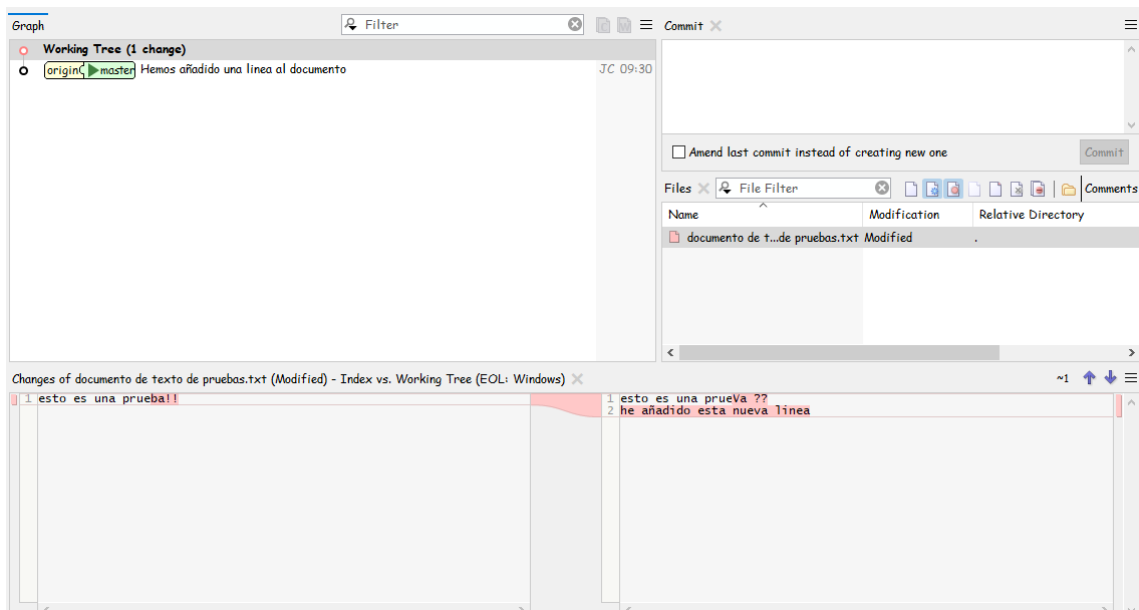
 **Hemos añadido una línea al documento**  
Jose Camacho authored 51 minutes ago 9ad83223

Name	Last commit	Last update
 documento de texto de pruebas.txt	Hemos añadido una línea al documento	51 minutes ago

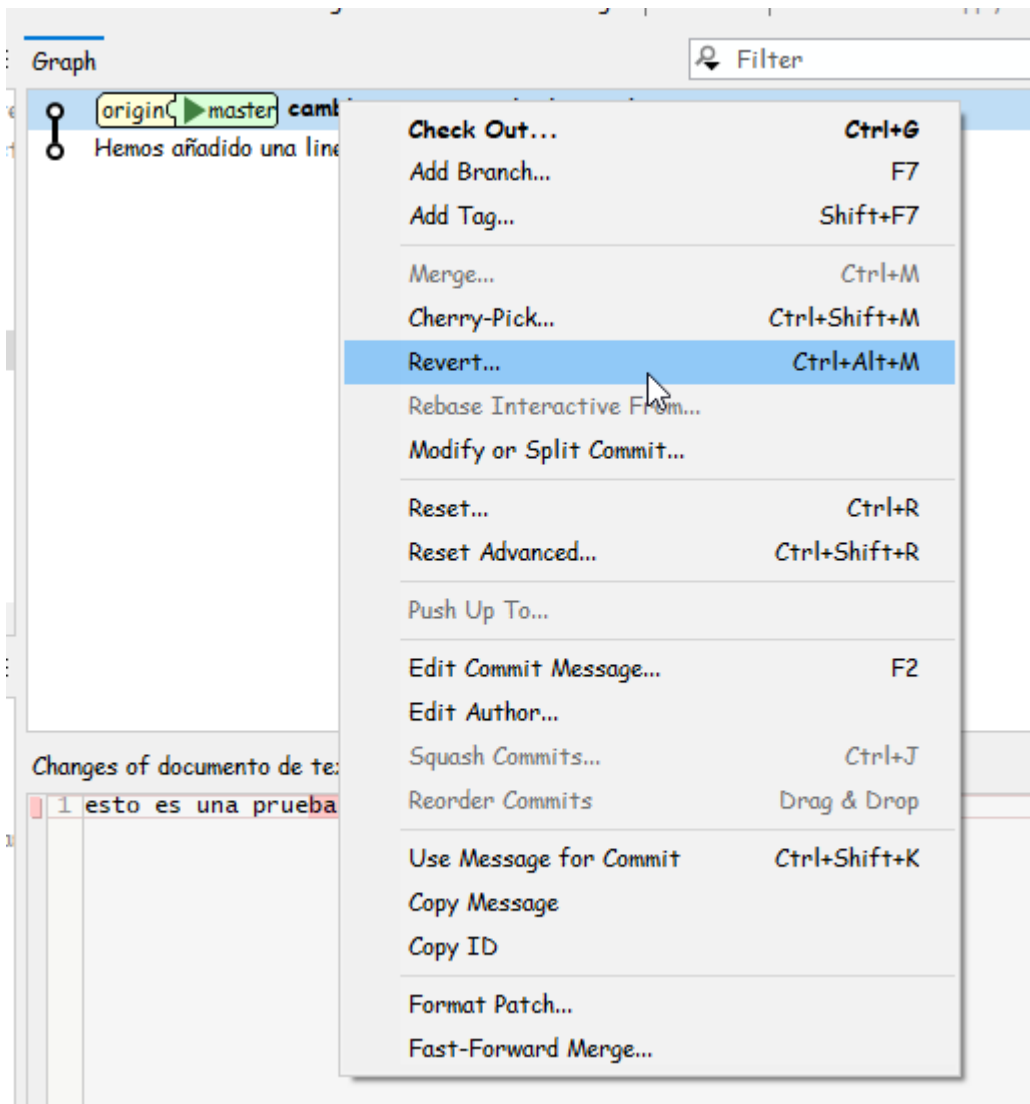
## 7. Revertir cambios

Puede que alguna vez nos equivoquemos a la hora de subir un archivo al repositorio, o que nos demos cuenta a las semanas de que el archivo que habíamos modificado tenía un error y queramos recuperar esa versión antigua.

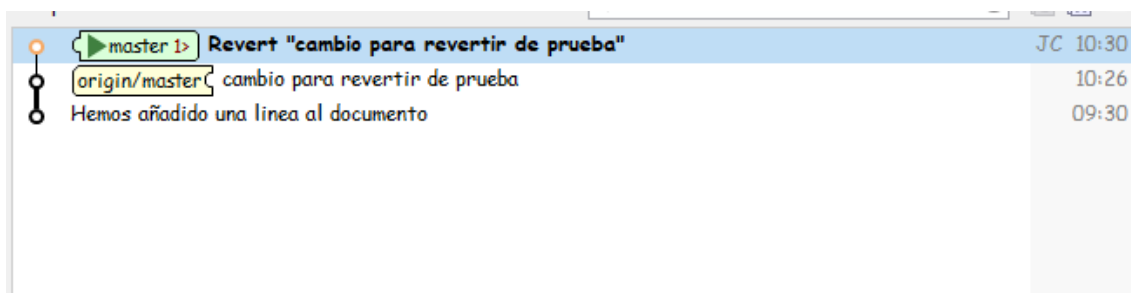
En el ejemplo vamos a añadir otra línea a nuestro documento de texto, y además, modificamos un poco la que ya tenemos. El cliente nos lo muestra así:



Hacemos commit y push para subir este cambio, y a continuación vamos a revertirlo. Hacemos click derecho sobre el último cambio, y clickamos sobre Revert...



Nos indicará que esto va a deshacer los cambios en este commit. Tenemos dos opciones, Revert para revertir y seguir trabajando, y más adelante guardar los cambios. Revert & Commit, para revertir y guardar los cambios, y a continuación seguir trabajando. En el ejemplo seleccionamos Revert & Commit.



Ahora ya tenemos el archivo revertido en nuestro local y podemos trabajar con él. No nos olvidemos que tanto si trabajamos con él como si no, este cambio se hace en local, por lo que tenemos que hacer un Commit (en el caso de que no lo hayamos hecho) y Push para subir esta versión del archivo a la última versión del código en el repositorio.