

**Manual de GIT y GitHub: Instalación y uso de comandos**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

Para comenzar con este manual, vamos a proceder a instalar GIT, el cuál nos da acceso a tener nuestros programas en la nube.

Primero, buscamos en nuestro navegador elegido, “GIT”, entraremos en el primer enlace mostrado.

<https://git-scm.com> ▾ [Traducir esta página](#)

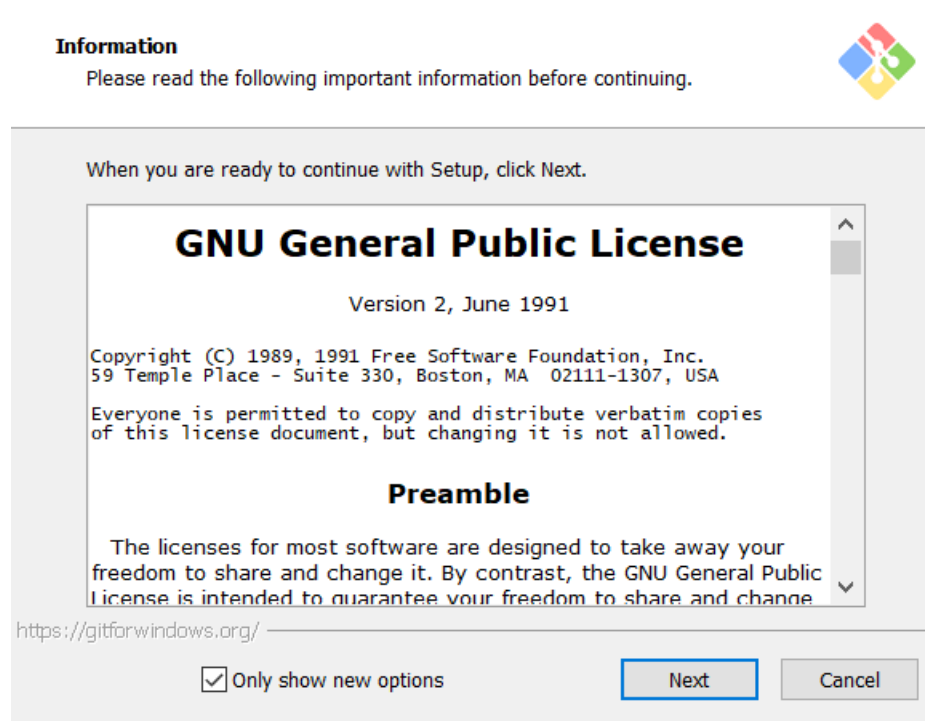
## Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

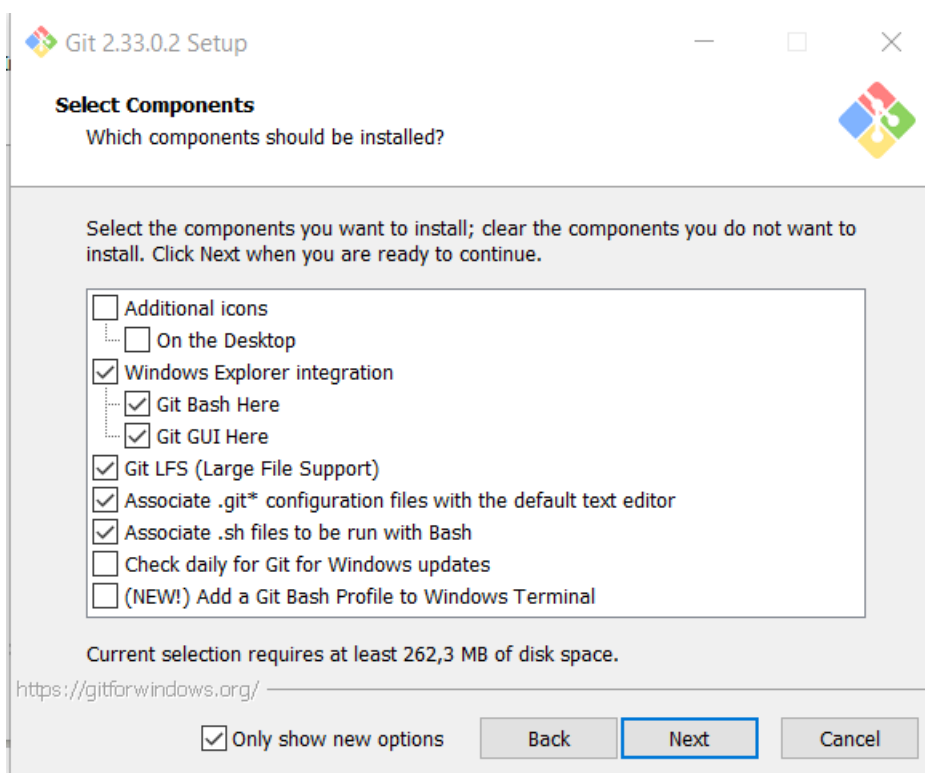
Nos aparecerá un recuadro, el cuál nos permitira descargar la versión más reciente de GIT para Windows:



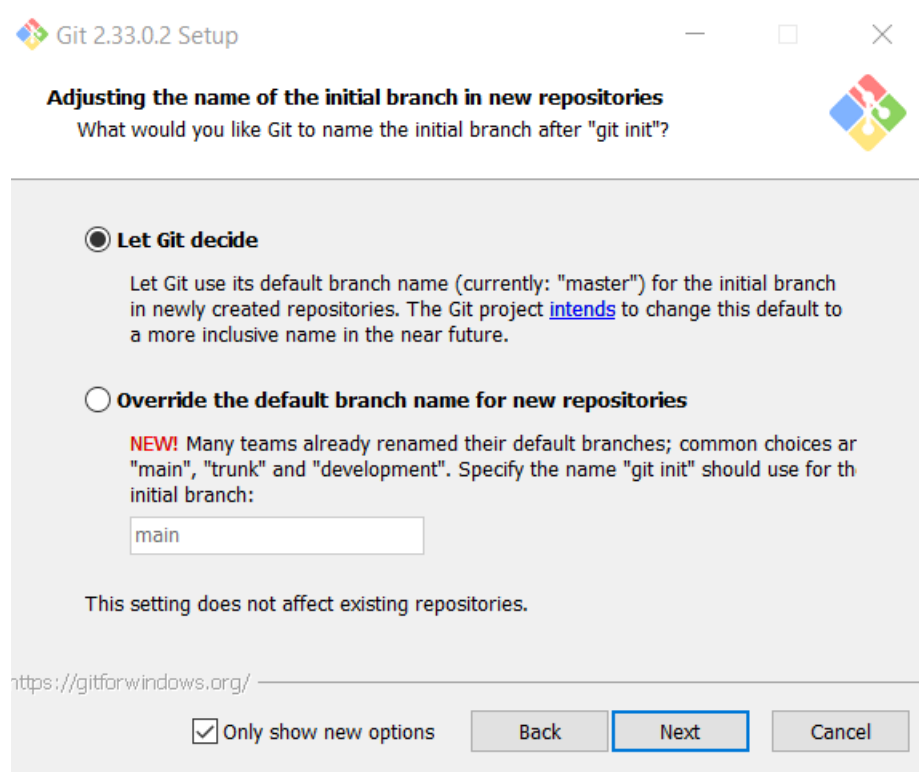
A continuación ejecutamos el archivo. Pinchamos en NEXT



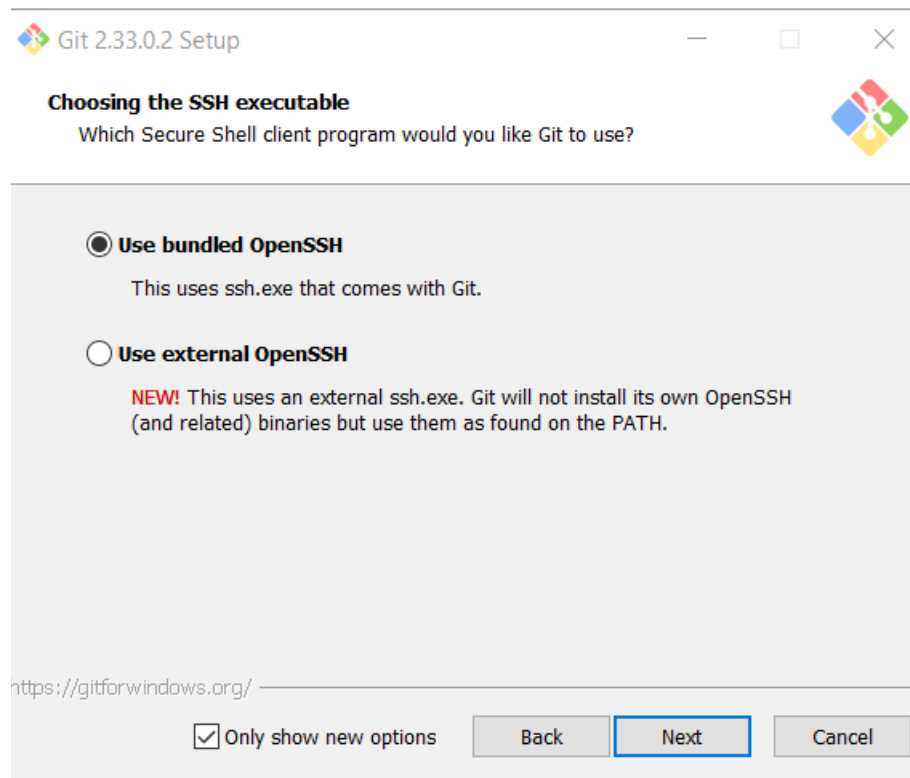
En esta pantalla nos pregunta que componentes queremos instalar, en mi caso son todos. Por lo que pincho en NEXT



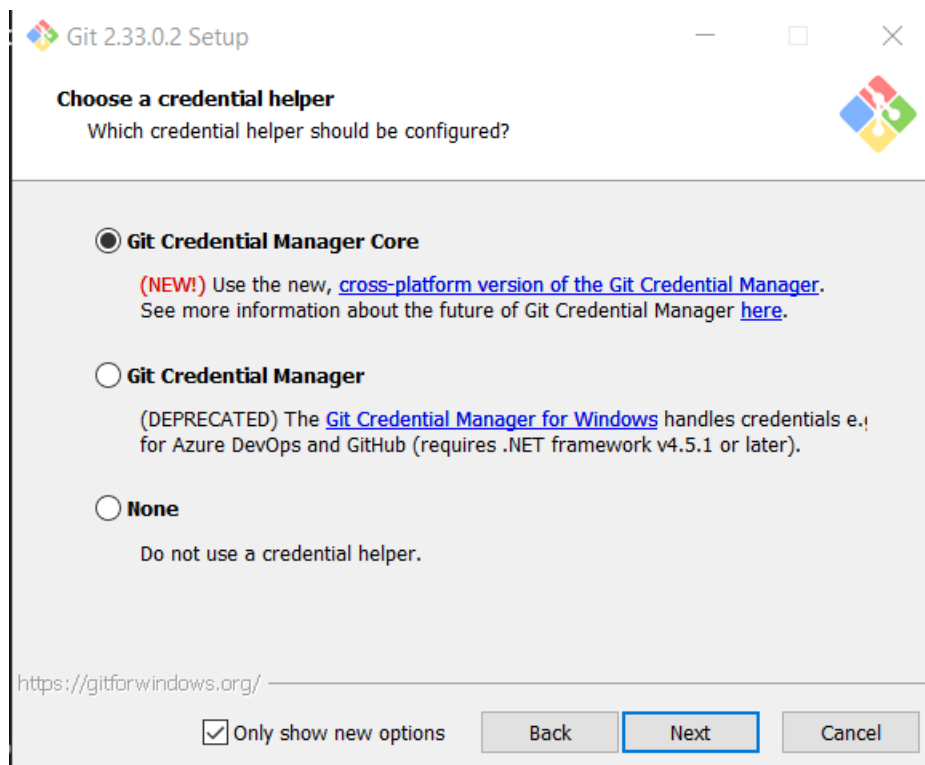
En este paso, pinchamos next.



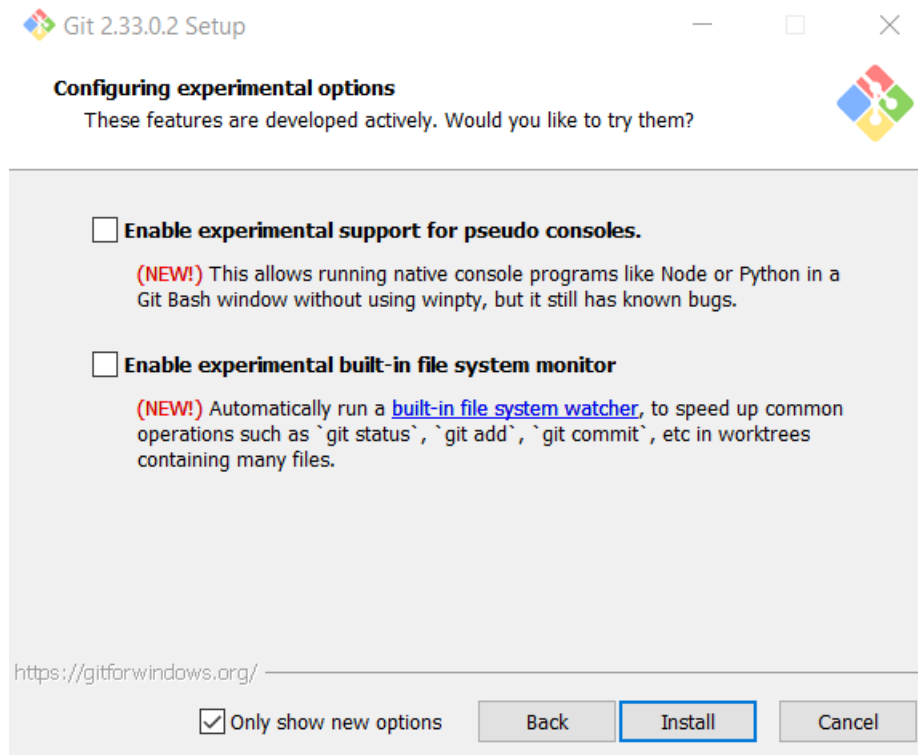
Seleccionamos NEXT.



Nos da a elegir entre asistentes de credenciales, lo dejamos por defecto.



Y por ultimo seleccionamos instalar.



## Comandos básicos GIT:

**-git init** creará un nuevo repositorio local GIT. El siguiente comando de Git creará un repositorio en el directorio actual:

*git init*

*Ejemplo:*

```
PS C:\Users\usuario> cd C:\Users\usuario\Documents\pruebal
PS C:\Users\usuario\Documents\pruebal> dir
```

Directorio: C:\Users\usuario\Documents\pruebal

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	20/09/2021 9:25	5	leer.txt

```
PS C:\Users\usuario\Documents\pruebal> git init
Initialized empty Git repository in C:/Users/usuario/Documents/pruebal/.git/
```

Como alternativa, puedes crear un repositorio dentro de un nuevo directorio especificando el nombre del proyecto:

*git init [nombre del proyecto]*

*Ejemplo:*

```
PS C:\Users\usuario\Documents\pruebal> git init
Initialized empty Git repository in C:/Users/usuario/Documents/pruebal/.git/
```

**-git add** se usa para agregar archivos al área de preparación.

*git add <temp.txt>*

*Ejemplo:*

```
PS C:\Users\usuario\Documents\pruebal> git add .
PS C:\Users\usuario\Documents\pruebal> git status
On branch master
```

No commits yet

**-git commit** creará una instantánea de los cambios y la guardará en el directorio git.

*git commit -m "El mensaje que acompaña al commit va aquí"*

Ten en cuenta que los cambios confirmados no llegarán al repositorio remoto.

Cuando se modifica, se hace un git status, en este paso se muestra un mensaje en el cuál se indica que se ha modificado, si se modifica se tiene que hacer un git add. para confirmar los cambios:

*Ejemplo:*

```
PS C:\Users\usuario\Documents\pruebal> git commit -m "first commit"
[master (root-commit) 18fa580] first commit
1 file changed, 1 insertion(+)
create mode 100644 leer.txt
```

**-git config** puede ser usado para establecer una configuración específica de usuario, como el email, nombre de usuario y tipo de formato, etc. Por ejemplo, el siguiente comando se usa para establecer un email:

*git config --global user.email tuemail@ejemplo.com*

La opción -global le dice a GIT que vas a usar ese correo electrónico para todos los repositorios locales.

Si quieres utilizar diferentes correos electrónicos para diferentes repositorios, usa el siguiente comando:

*git config --local user.email tuemail@ejemplo.com*

*Ejemplo:*

```
PS C:\Users\usuario\Documents\pruebal> git config --global user.email "rcarmorc@g.educaand.es"
PS C:\Users\usuario\Documents\pruebal> git config --global user.name "rafacarretero"
```

**-git status** muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser preparados o confirmados.

*git status*

*Ejemplo:*

```
PS C:\Users\usuario\Documents\pruebal> git add .
PS C:\Users\usuario\Documents\pruebal> git status
On branch master
```

No commits yet

**-git push** se usa para enviar confirmaciones locales a la rama maestra del repositorio remoto. Aquí está la estructura básica del código:

*git push origin <master>*

Reemplaza <master> con la rama en la que quieres enviar los cambios cuando no quieras enviarlos a la rama maestra.

*Ejemplo:*

```
PS C:\Users\usuario\Documents\pruebal> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 216 bytes | 216.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/rafacarretero/pruebal.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

**-git remote** te permite ver todos los repositorios remotos. El siguiente comando listará todas las conexiones junto con sus URLs:

*git remote -v*

-Para conectar el repositorio local a un servidor remoto, usa este comando:

*git remote add origin <host-or-remoteURL>*

Por otro lado, el siguiente comando borrará una conexión a un repositorio remoto especificado:

*git remote <nombre-del-repositorio>*

*Ejemplo:*

```
PS C:\Users\usuario\Documents\pruebal> git remote add origin
https://github.com/rafacarretero/pruebal.git
```

**-git branch** se usa para listar, crear o borrar ramas. Por ejemplo, si quieres listar todas las ramas presentes en el repositorio, el comando debería verse así:

*git branch*

Si quieres borrar una rama, usa:

*git branch -d <branch-name>*

*Ejemplo:*

```
PS C:\Users\usuario\Documents\pruebal> git branch -M main
```