

¿Qué es Git?

Git es un software de **control de versiones distribuido** (Es un software libre).

El propósito de esta Git, es gestionar/registrarse los cambios de los diversos cambios que se realizan en los archivos de una aplicación o software.

Git está diseñado para coordinar el trabajo de diferentes personas (Desarrolladores) sobre un conjunto de archivos compartidos, manteniendo la eficiencia y fiabilidad de las versiones del software.

La principal característica de Git, es que es un control de versiones distribuido, al ser distribuido permite trabajar a varios programadores en un proyecto (software) sin necesidad de compartir la misma red.



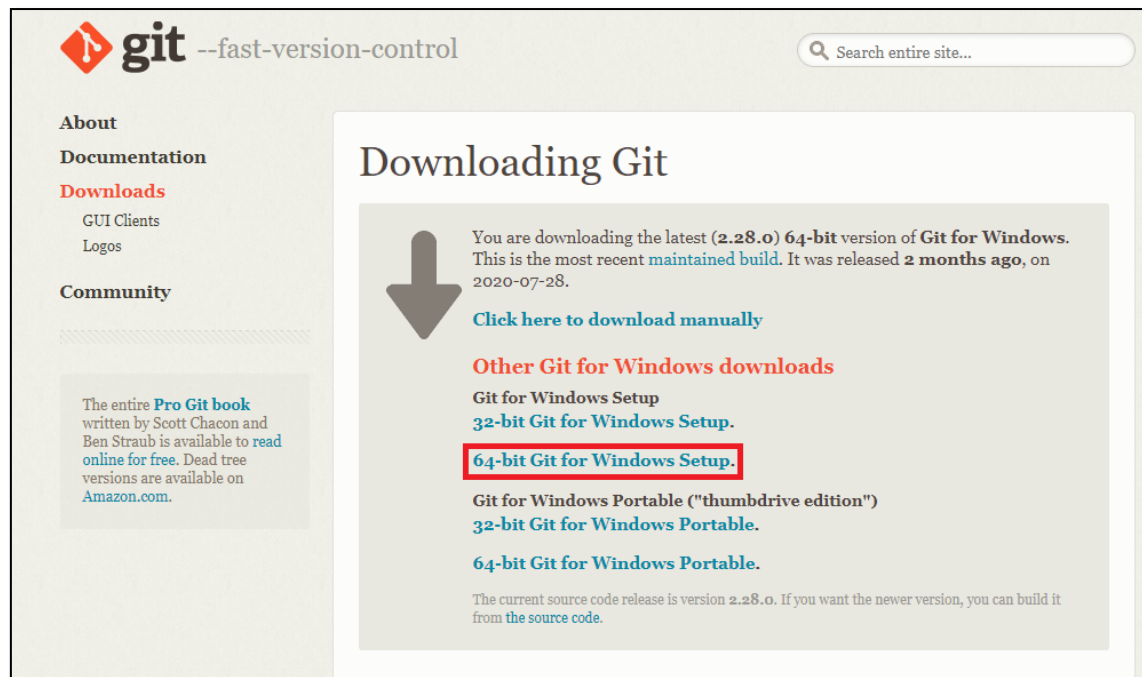
Instalación de Git

A continuación, vamos a redactar el proceso de instalación de Git en Windows.

El primer paso es descargar el instalador de Git correspondiente (Depende de nuestra arquitectura de S.O) (Recomiendo instalar la versión setup). En nuestro caso vamos a instalar la versión 2.28.0.

La descarga se realizará desde su página web oficial (Es gratuito), el enlace de descarga oficial es el siguiente:

<https://git-scm.com/download/win>



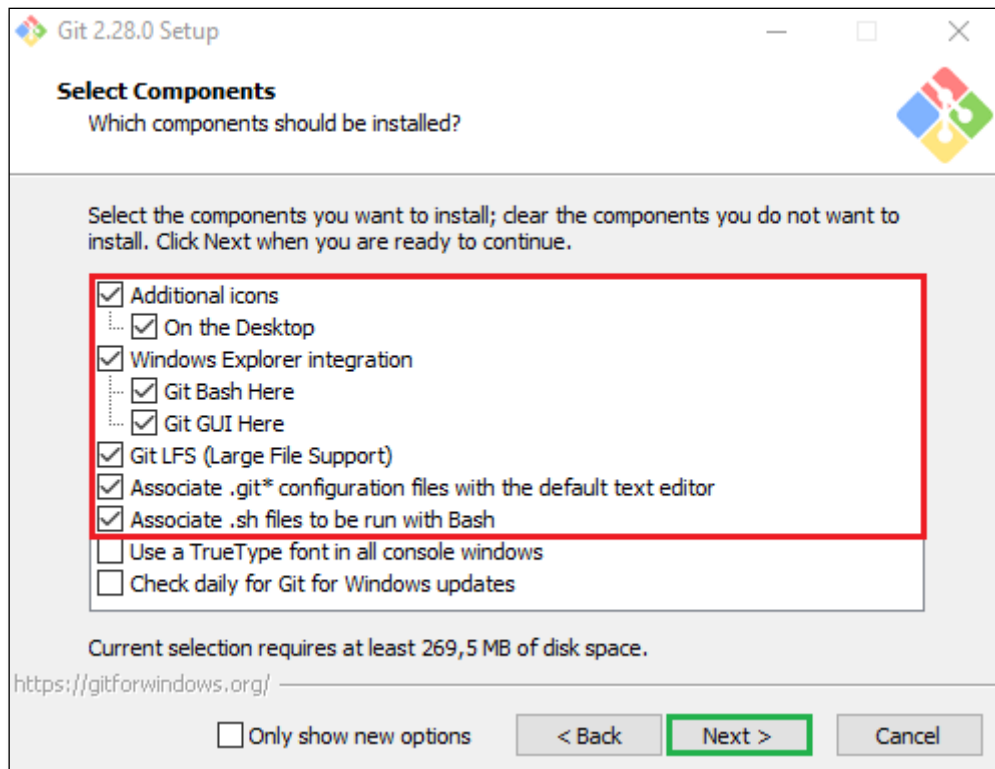
Una vez descargado, localizaremos el archivo de instalación, predeterminadamente se localizará en la carpeta "Descargas", una vez localizado realizaremos doble clic izquierdo para ejecutarlo.

* Nos solicitará permisos de administrador para ejecutarlo, deberemos aceptar.

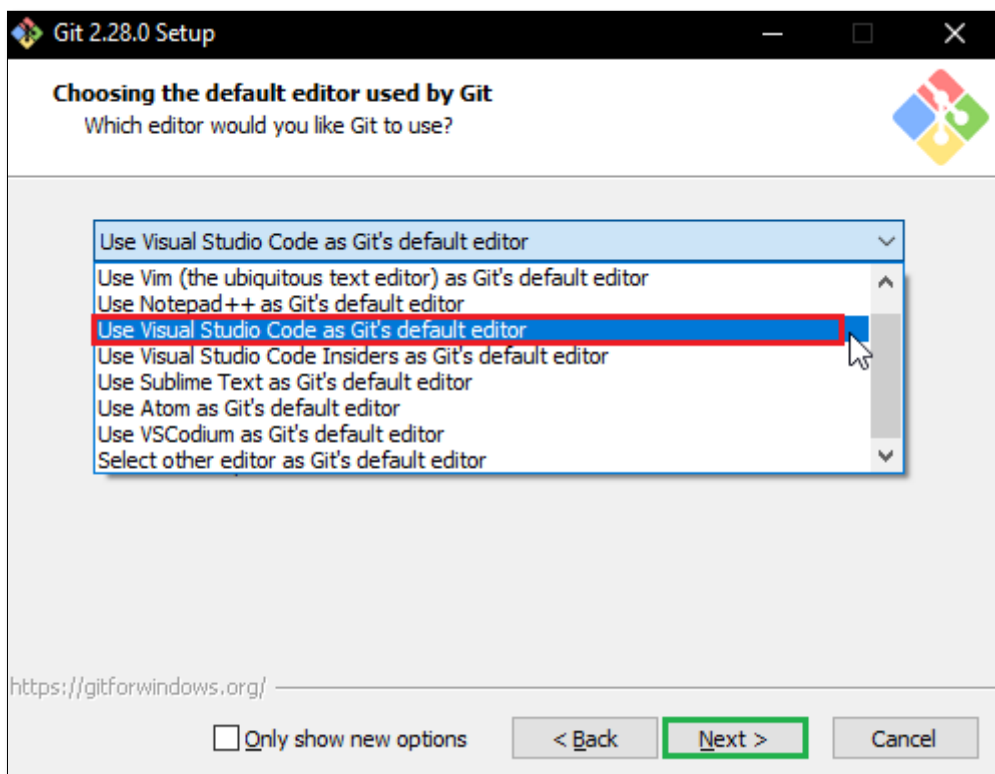
El primer paso del instalador, nos muestra la licencia que utiliza Git (GNU GPL), tendremos que aceptarla para proceder con la instalación.



A continuación, el instalador nos solicitará que marquemos **los componentes que queremos instalar**, en nuestro caso, seleccionaremos la predeterminadas.



El tercer paso del instalador, es **seleccionar el editor de código predeterminado**, hay diferentes opciones, en mi caso seleccionar "*Visual Studio Code*", pero podéis seleccionar cualquiera de las múltiples opciones que nos ofrece.



Una vez realizado el anterior paso, en el siguiente, deberemos **ajustar la configuración de entorno**, esto consiste en seleccionar como queremos trabajar con Git, en nuestro caso, seleccionaremos la opción predeterminada.

Las opciones son las siguientes:

Use git from the bash only:

Esta es la opción más segura porque no modifica el path, con esta opción solo podremos ejecutar Git desde la consola de Git.

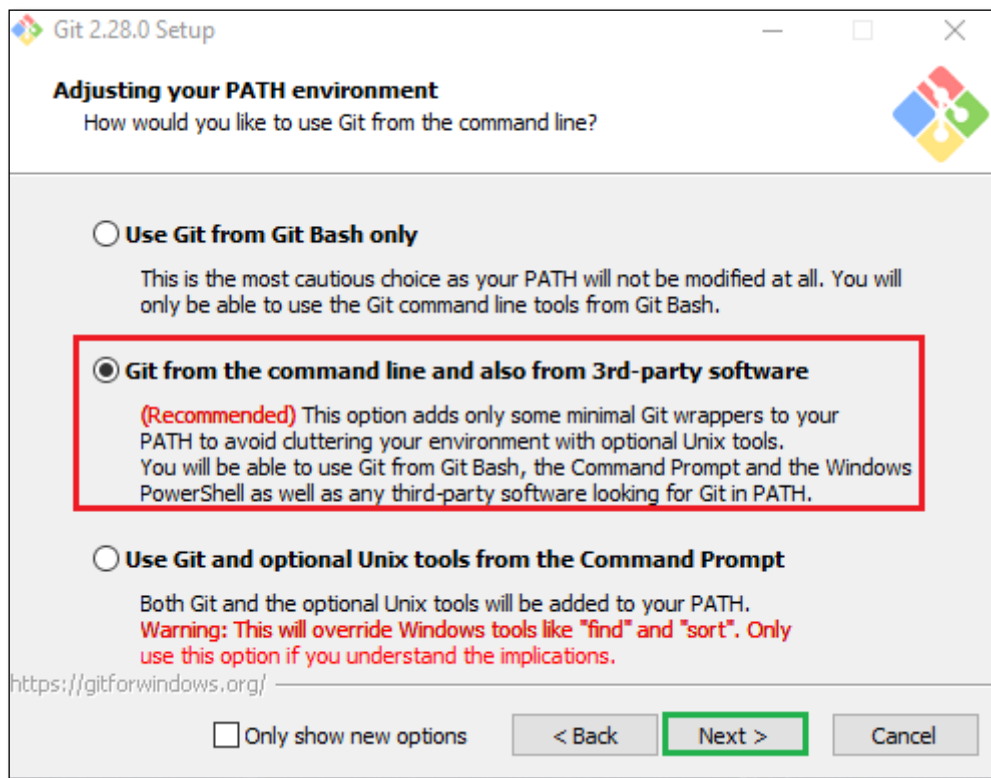
Git from the command line and also from 3rd-party software (predeterminada):

Esta opción añade los empaquetadores (Wrappers) de Git a nuestro Path, para evitar el desorden del entorno, con esta opción podremos que ejecutar Git, desde la consola de Git, CMD, Powershell y software de terceros.

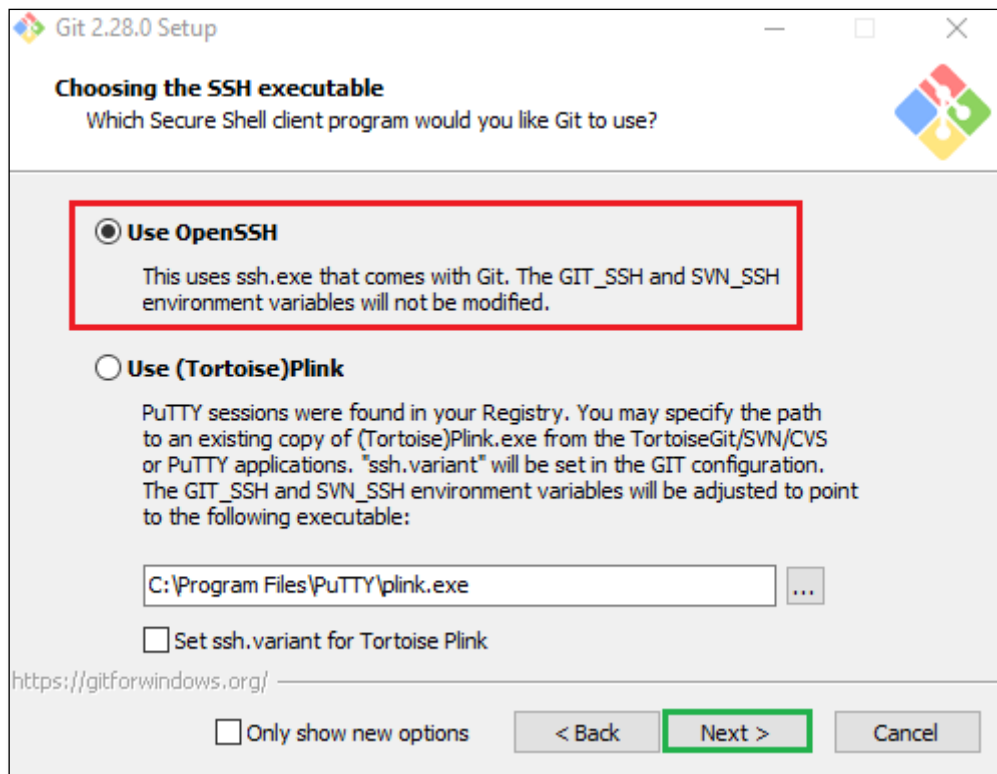
Use Git and optional Unix tools from the command prompt:

Esta opción agregará a nuestro Path tanto git como las herramientas opcionales Unix, por lo que podremos Usar git y las herramientas opcionales desde CMD

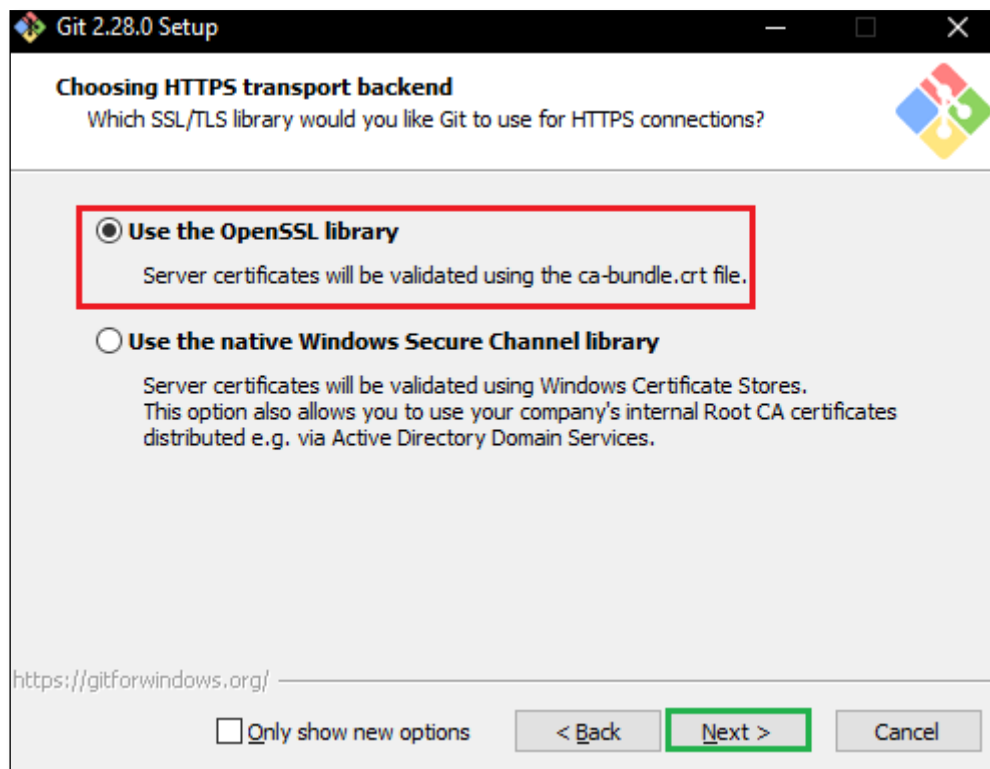
Atención: Esta opción anulará algunos comandos de Windows, por ejemplo, Find o Sort, así que deberemos tener cuidado, y tener en cuenta los comandos que utilizamos



A continuación, deberemos seleccionar el programa de seguridad de la Shell del cliente (Nosotros), en nuestro caso seleccionaremos la opción **“Use OpenSSH**, debido a que es el programa más común, y más seguro (No modifica las variables del sistema)

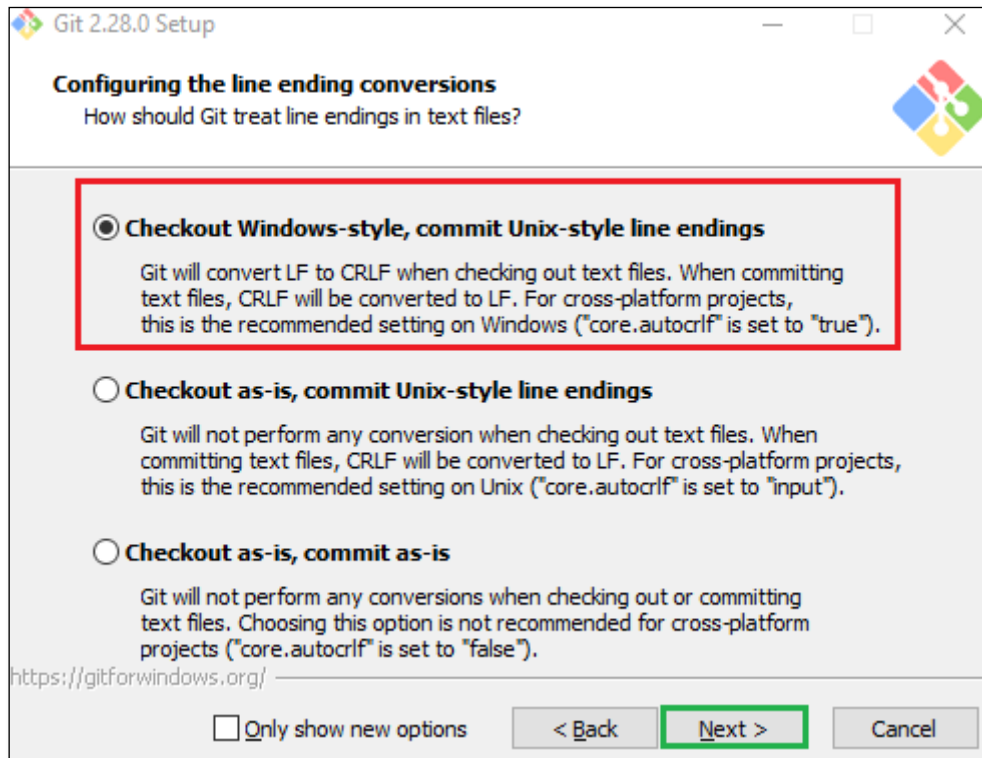


El siguiente paso, es seleccionar la librería (Conjunto de implementaciones) que usará Git para las conexiones Https con el *backend* (Servidor), en nuestro caso utilizaremos *OpenSSL*, puesto que es libre, y es muy común su uso.

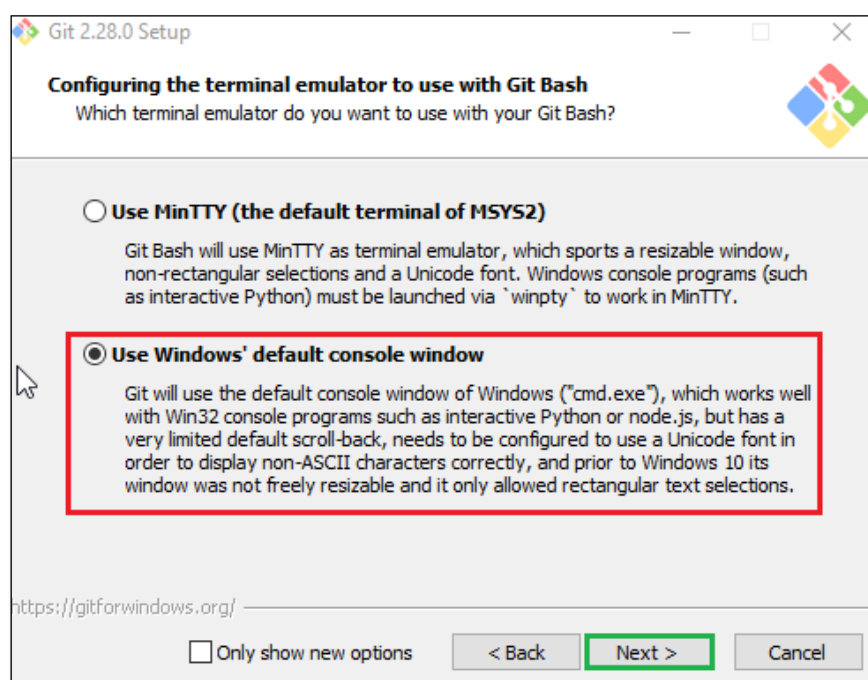


A continuación, deberemos **seleccionar los finales de línea**, de los archivos que interpretará Git, en nuestro caso como estamos en Windows, seleccionaremos la opción **“Checkout Windows-Style, commit Unix-Style Line endings”**.

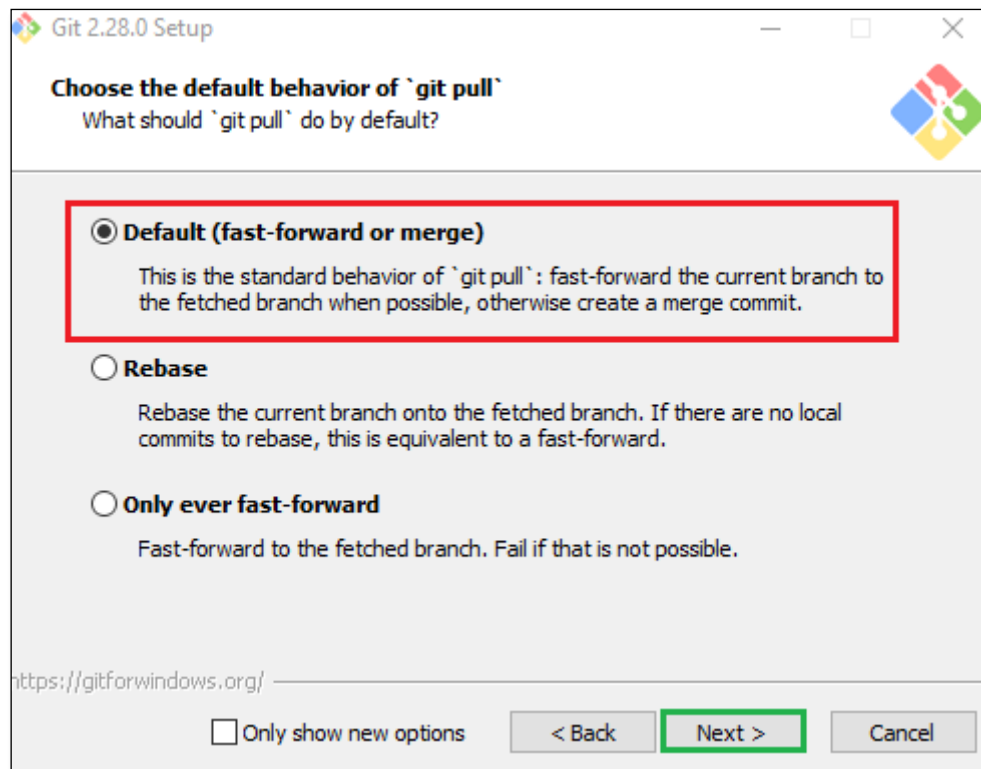
Seleccionaremos esta opción, puesto que como estamos trabajando en Windows, los archivos que creemos serán del tipo LF, esta opción convierte los archivos LF en CRLF, que es un tipo de archivo válido para multiplataformas. El resto de opciones no realizan conversiones.



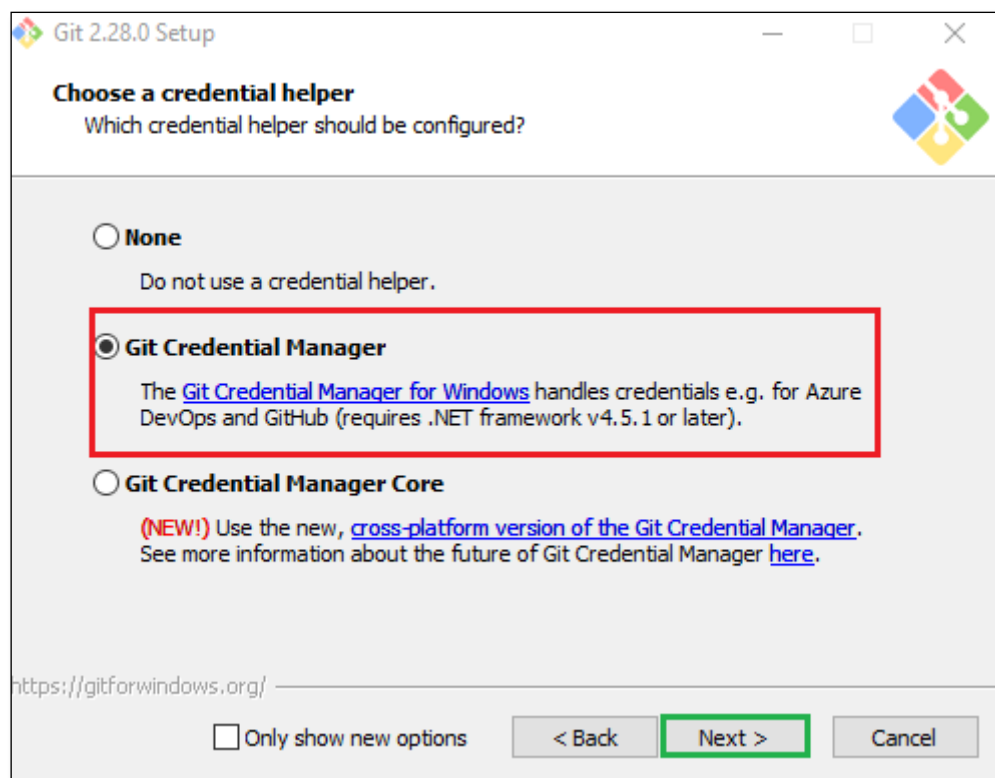
En el siguiente paso del instalador, deberemos seleccionar la aplicación con la que vamos a emular el bash de Git, en este caso como estamos trabajando y realizando la instalación en Windows, la opción más cómoda es seleccionar la línea de comandos de Windows, para ello seleccionaremos la opción **“Use Windows default console window”**.



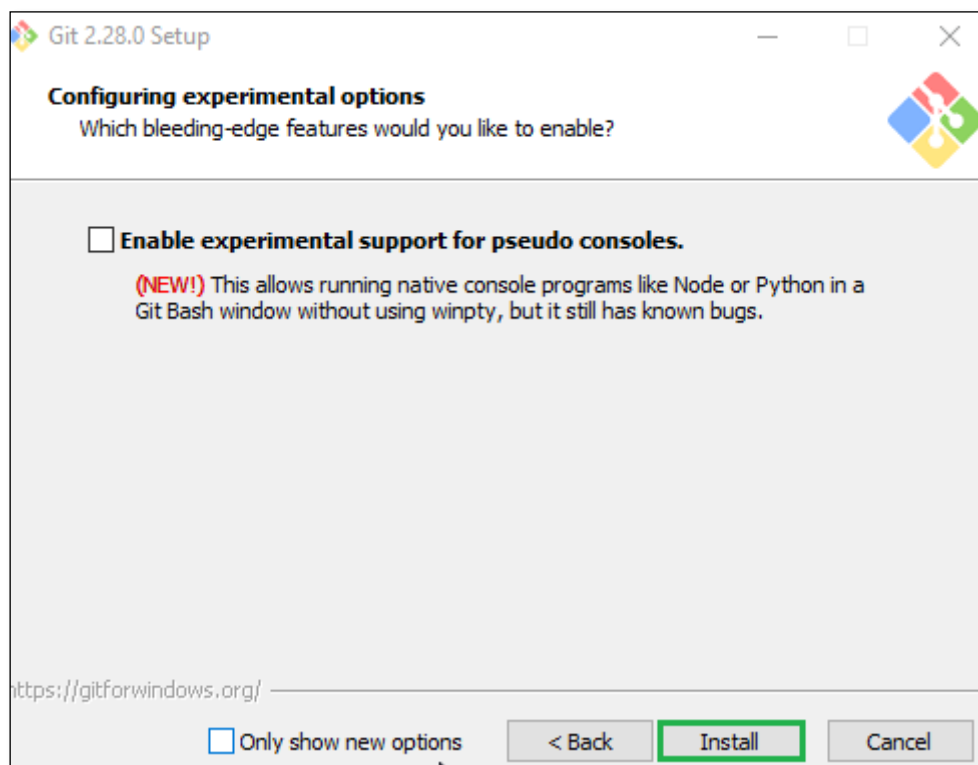
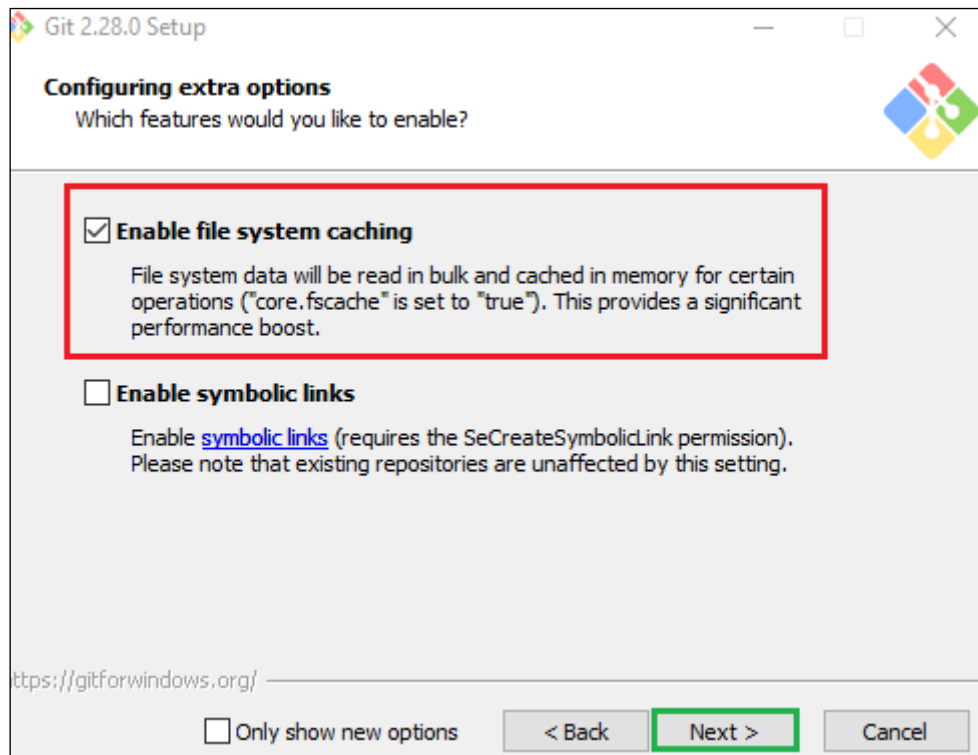
A continuación, deberes seleccionar que acción realizará el comando "*git pull*" (Comando muy utilizado en los proyectos Git), en nuestro caso seleccionaremos la opción por defecto, esta opción la seleccionamos para evitar que se unan los commit.



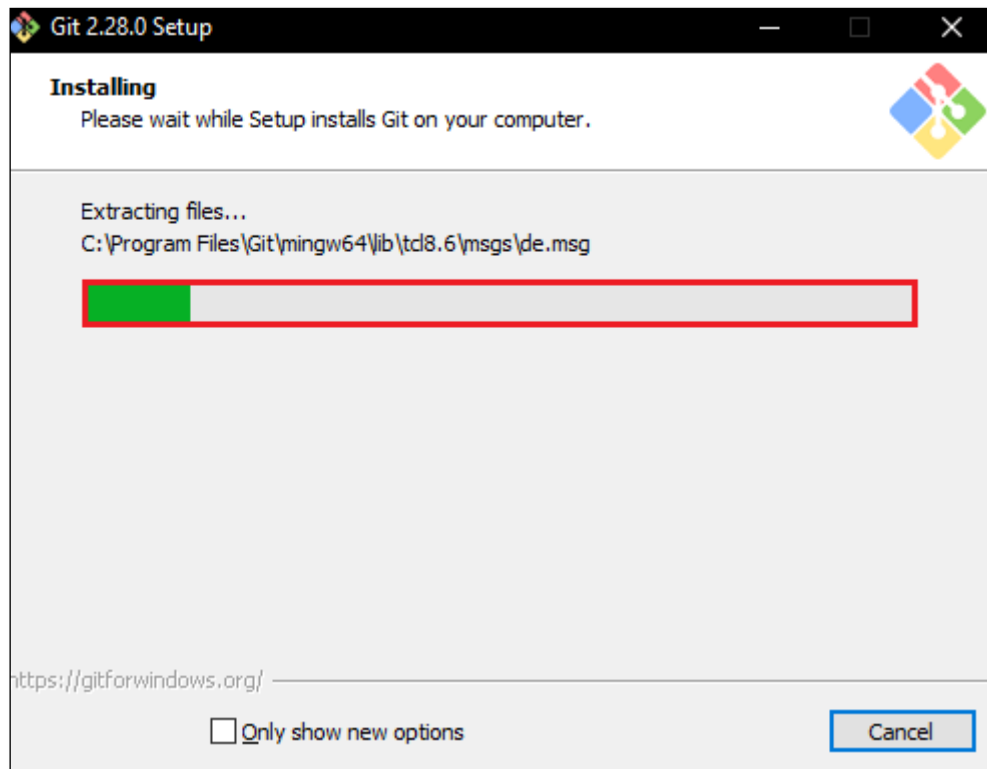
Deberemos seleccionar o no seleccionar ayudante de credenciales para facilitarnos el iniciar sesión, seleccionaremos la opción por defecto, por que es más estable que la otra opción, que es más nueva y por la tanto más propensa a fallos.



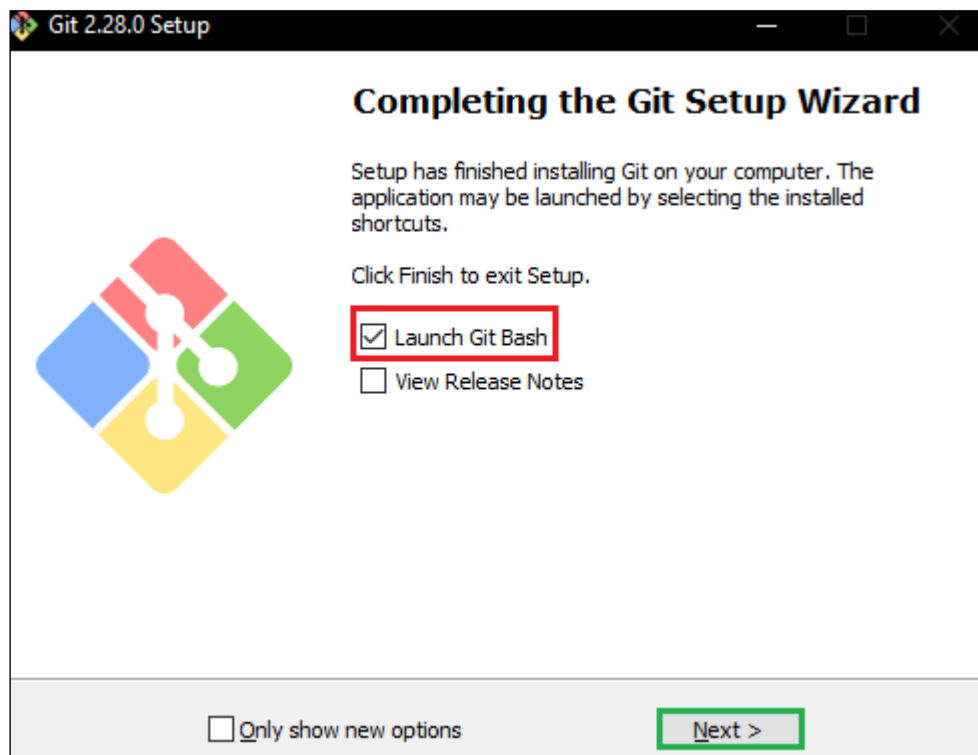
Para finalizar la instalación de Git, debemos seleccionar si queremos instalar opciones extra y las opciones experimentales, en nuestro caso hemos habilitado solo la opción **“Enable file system caching”**, que habilita el guardado de archivos en la cache para una mayor rapidez en el trabajo, el resto de opciones no las hemos seleccionado para evitar posibles fallos

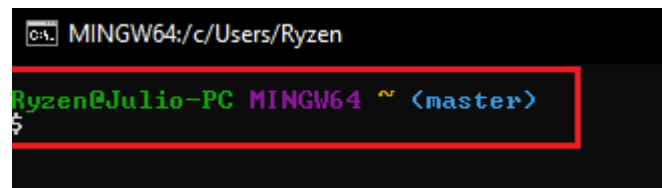


A continuación, comenzará el proceso de instalación, deberemos esperar hasta que la barra de progreso se complete, y nos muestre la ventana de finalización de la instalación.



En la ventana de finalización de la instalación, marcaremos la opción “*Launch Git bash*”, para comprobar que la instalación se ha realizado correctamente.





A screenshot of a terminal window with a black background. The title bar at the top reads "MINGW64:/c/Users/Ryzen". The terminal content shows a Git commit message in a multi-line format: "Ryzen@Julio-PC" in green, "MINGW64" in purple, a tilde symbol "~" in yellow, and "<master>" in blue. A red rectangular box highlights the commit message text.

```
C:\> MINGW64:/c/Users/Ryzen  
Ryzen@Julio-PC MINGW64 ~ <master>  
$
```