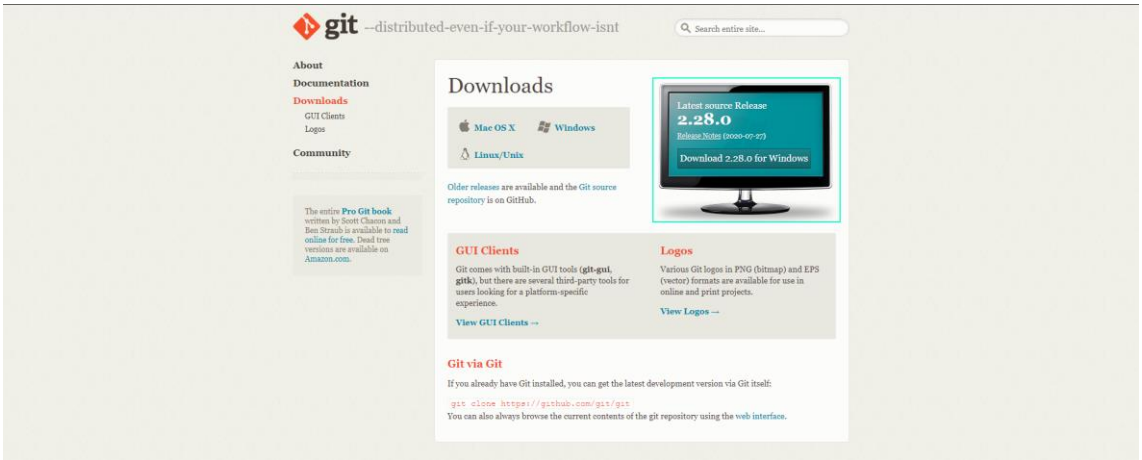
Descarga e instalación de Git

Procederemos a la descarga e instalación de Git.

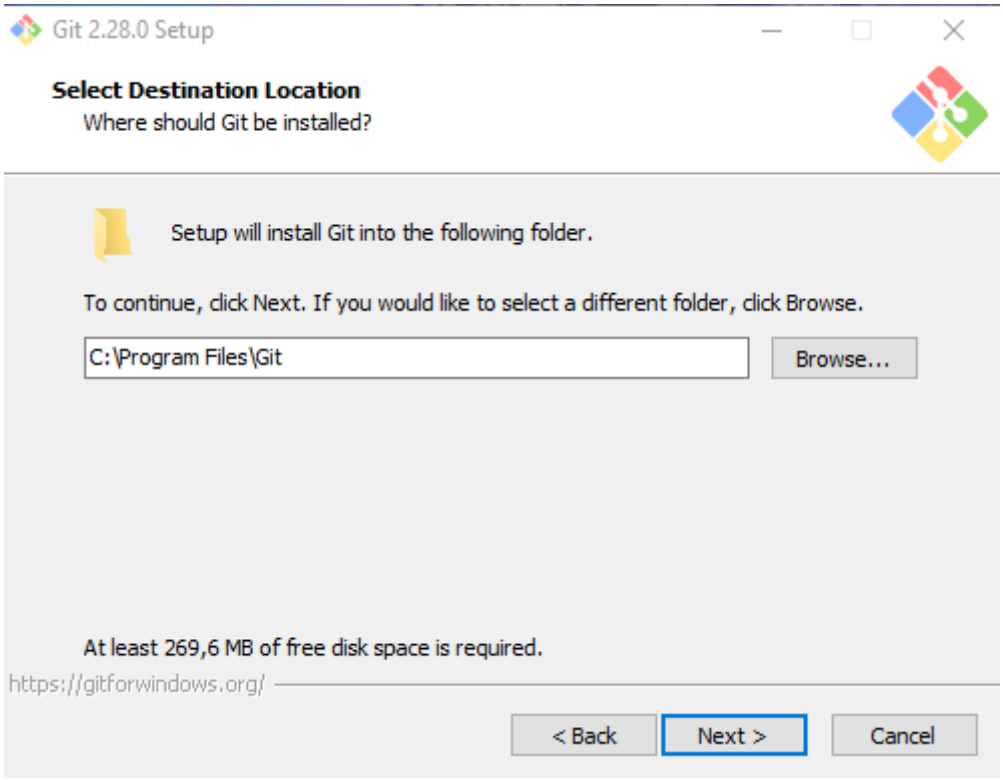
- Enlace de descarga: https://git-scm.com/downloads

Al igual que con Mongo, accederemos al enlace proporcionado arriba y descargaremos la versión más reciente.

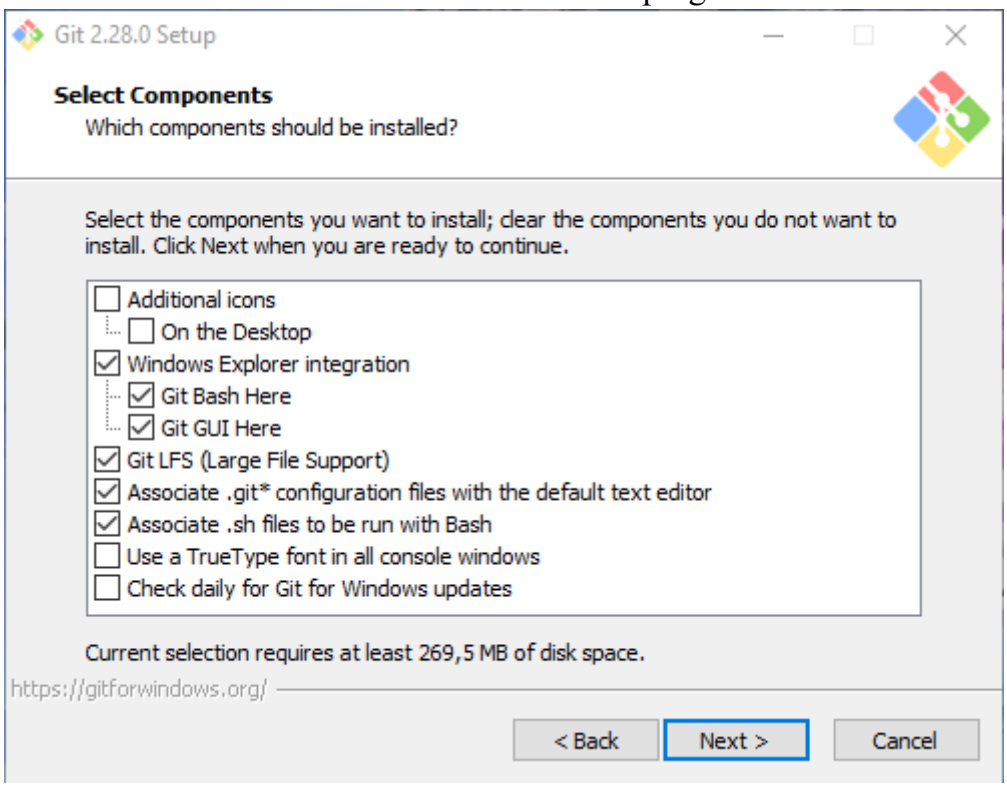


Abriremos el instalador que nos hemos descargado y comenzaremos el proceso de instalación aceptando la licencia de uso.

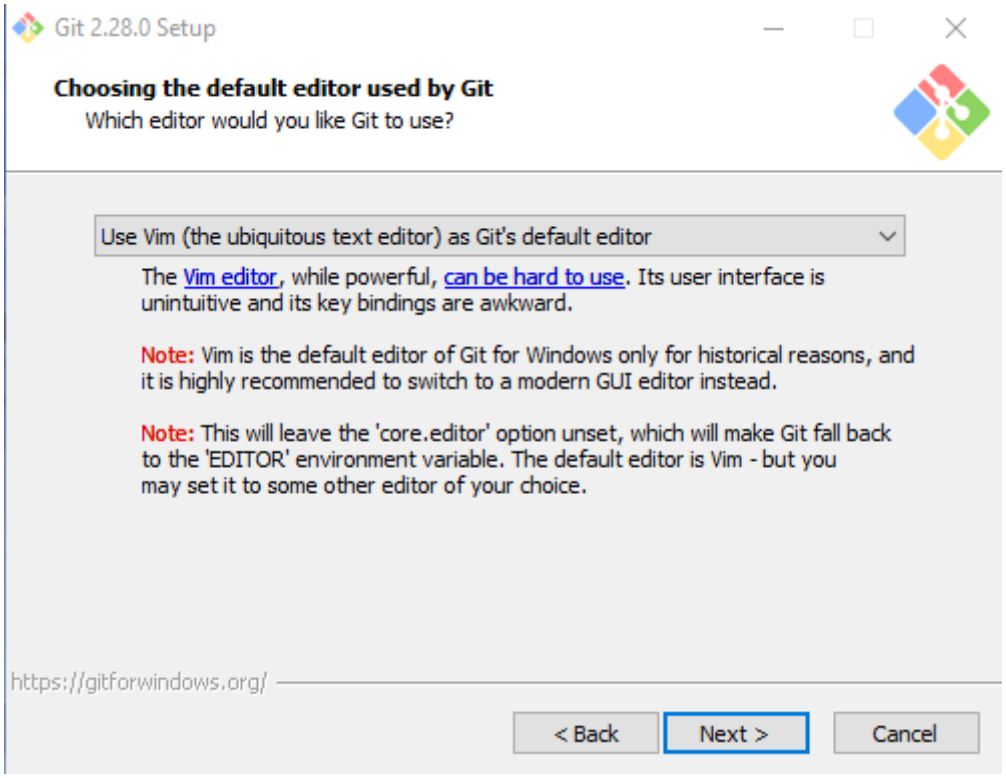
Dejaremos el directorio de instalación por defecto



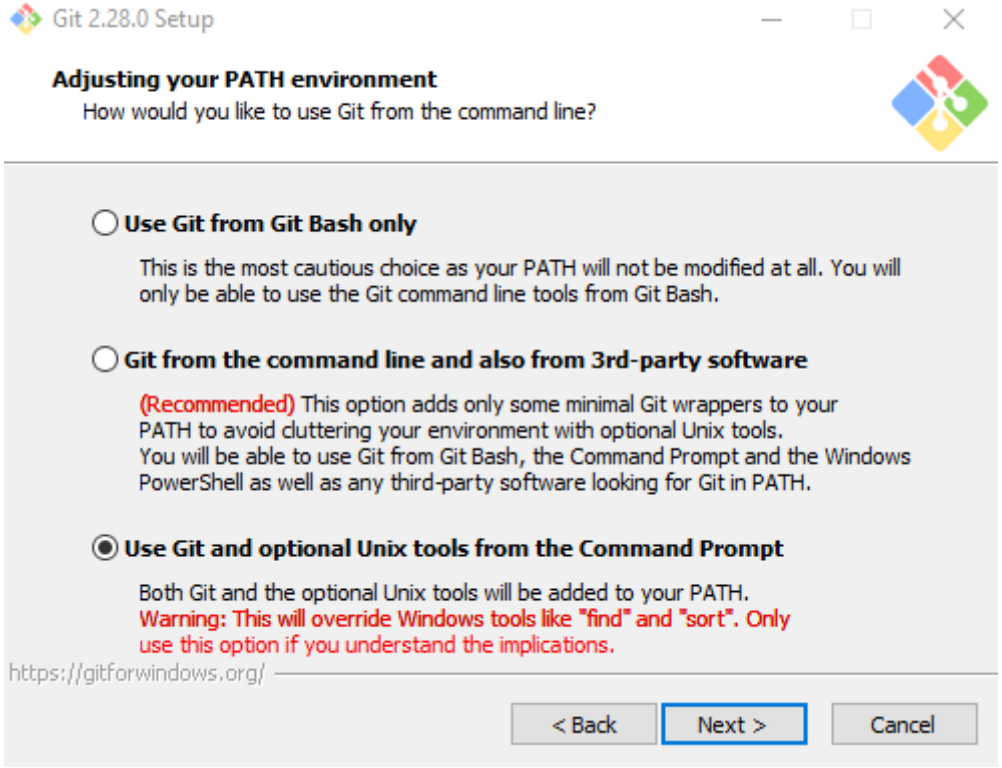
Tras esto, se nos abrirá la siguiente ventana, dejaremos todo tal y como se ve en la imagen que es la configuración por defecto, esto hará que se instale un Bash en Windows lo cual hará más cómodo el uso del programa. Le damos a “Next”.



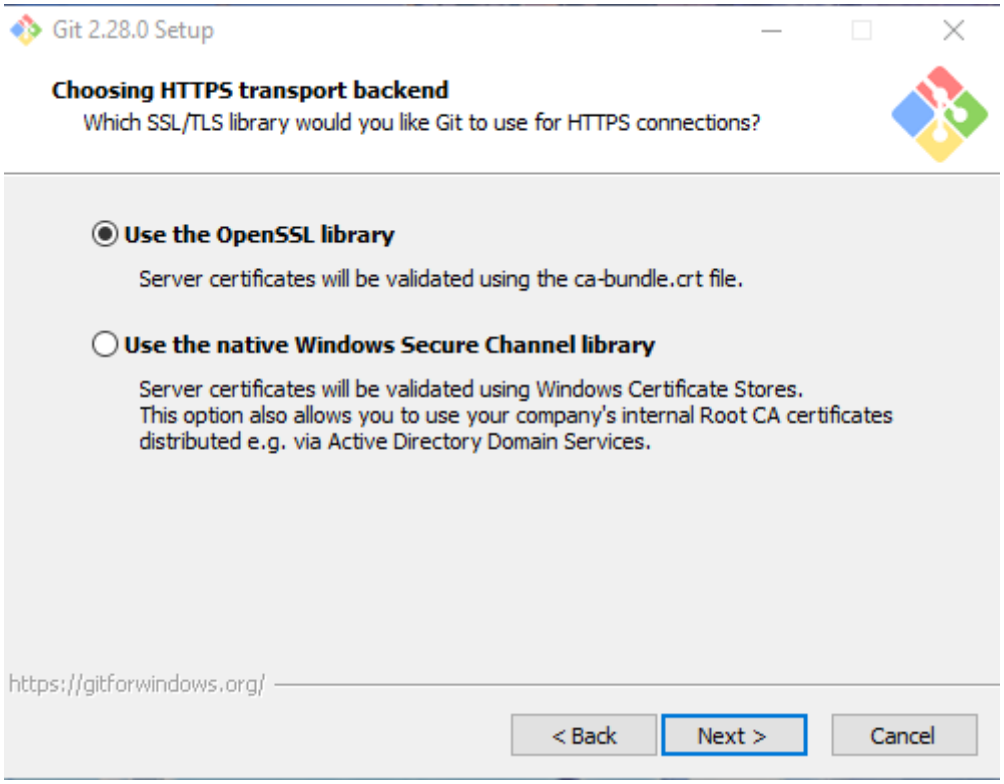
En la siguiente ventana dejaremos la configuración por defecto otra vez puesto que usaremos el editor que más cómodo nos parezca. Pulsamos “Next”.

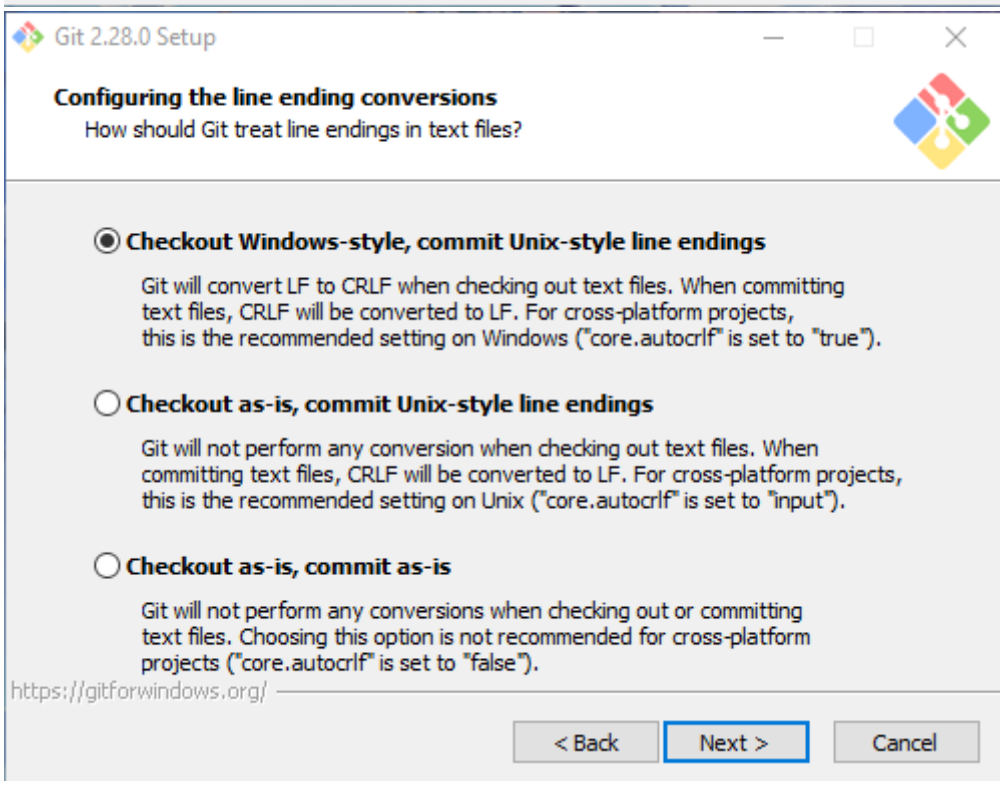


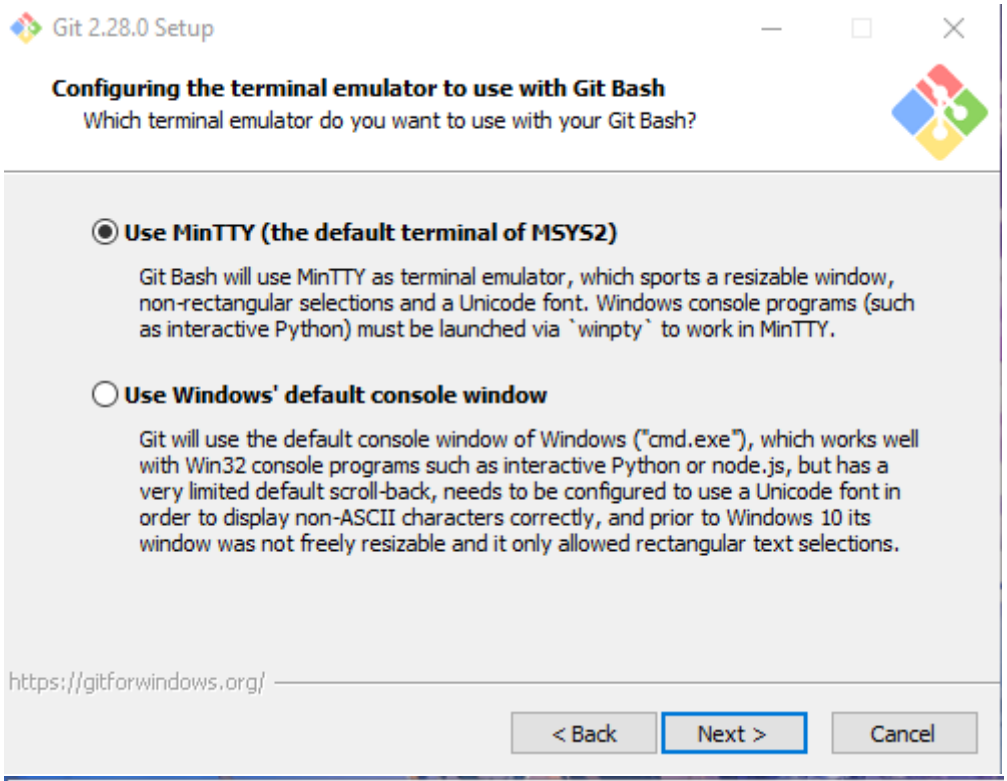
En esta ventana, seleccionaremos la última opción para que Git se nos instale en nuestro Path por defecto, lo cual hace más cómodo su instalación ya que no tenemos que ponerlo manualmente.

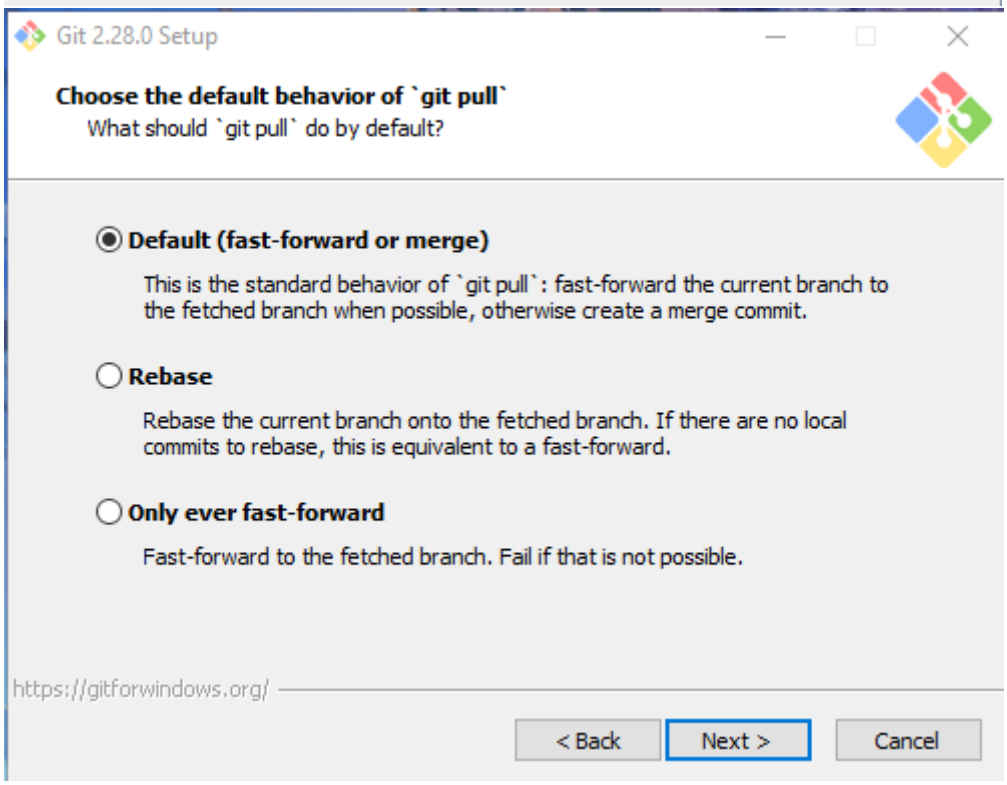


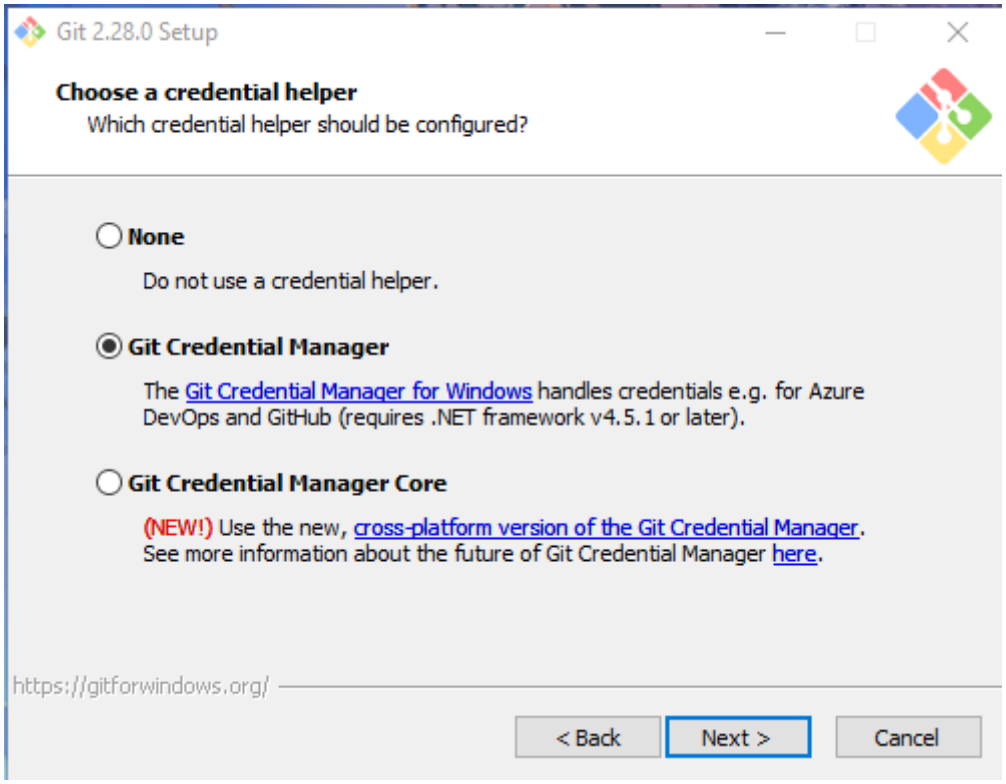
Y ya para terminar, en las siguientes ventanas dejaremos todo por defecto, puesto que es para instalar el Bash de Git en nuestro sistema.

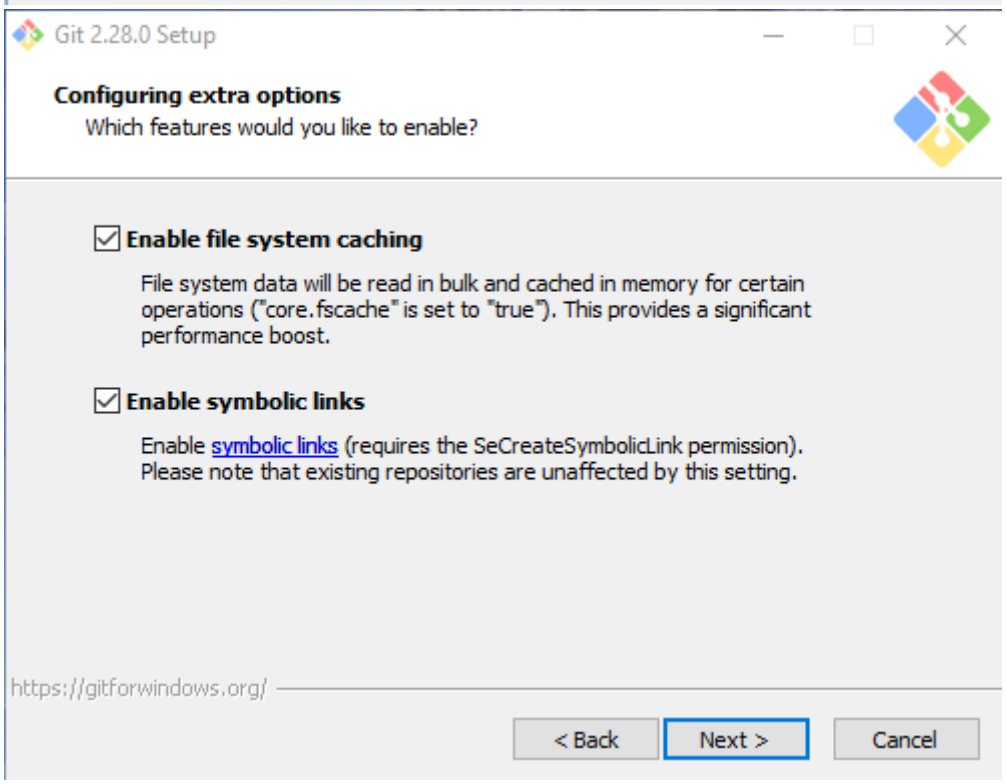

Git 2.28.0 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates
distributed e.g. via Active Directory Domain Services.
https://gitforwindows.org/
< Back
Next >
Cancel

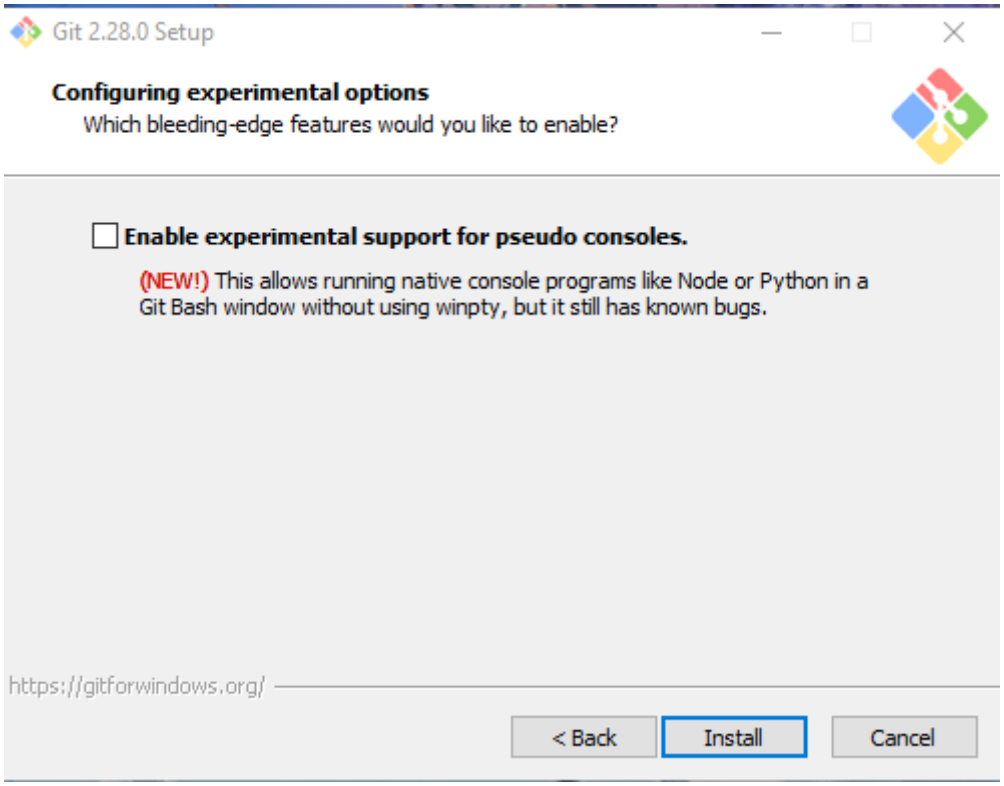
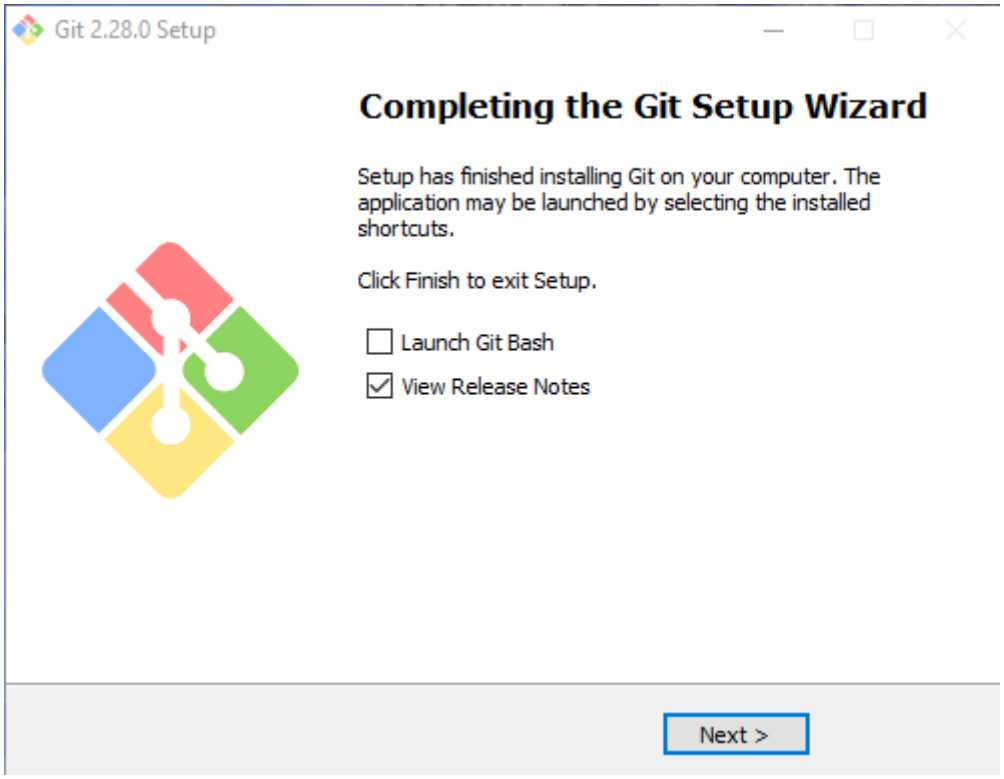

Git 2.28.0 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing
text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When
committing text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing
text files. Choosing this option is not recommended for cross-platform
projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
< Back
Next >
Cancel


Git 2.28.0 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
https://gitforwindows.org/
< Back
Next >
Cancel


Git 2.28.0 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Default (fast-forward or merge)
This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
Rebase
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.
Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible.
https://gitforwindows.org/
< Back
Next >
Cancel


Git 2.28.0 Setup
Choose a credential helper
Which credential helper should be configured?
None
Do not use a credential helper.
Git Credential Manager
The Git Credential Manager for Windows handles credentials e.g. for Azure DevOps and GitHub (requires .NET framework v4.5.1 or later).
Git Credential Manager Core
(NEW!) Use the new, cross-platform version of the Git Credential Manager.
See more information about the future of Git Credential Manager here.
https://gitforwindows.org/
< Back
Next >
Cancel


Git 2.28.0 Setup
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission).
Please note that existing repositories are unaffected by this setting.
https://gitforwindows.org/
< Back
Next >
Cancel



Pulsamos “Install” en esta ventana y habremos terminado la instalación de Git.