UNIVERSIDAD DON BOSCO
LENGUAJES INTERPRETADOS EN EL CLIENTE
CICLO 2- 2022

TALLER #1

GT: 02

INTEGRANTES:

Apellidos	Nombres	Carnet
Alvarenga Claros	Carlos Alexander	AC211104
Roberto Rodrigo	López Alvarado	LA203000
Guevara Landaverde	Hugo Fernando	(GL211421)
Zaldaña Álvarez	Francisco Javier	(ZA210751)

DOCENTE: Ing. Delmy Azucena Majano Menjivar

FECHA DE ENTREGA: 12 de septiembre de 2022

Importancia del uso de sistemas de control de versiones

Para hablar acerca de la importancia del uso de sistemas de control de versiones es importante el aprender un poco más acerca de que es un sistema de control de versiones. Para los proveedores de software, los cambios en un producto de software que no modifican el producto subyacente, sino que agregan funcionalidad y solucionan problemas, se denominan nuevas versiones de software. Las actualizaciones de software son parte de cualquier ciclo de vida del software. Con el tiempo, se lanzan nuevas funciones, los desarrolladores corrigen errores y las extensiones de software evolucionan con emocionantes actualizaciones y nuevas versiones. Por lo tanto, el control de versiones de software es una parte integral de la producción.

El control de versiones de software es importante porque ayuda a los usuarios y proveedores de software a realizar un seguimiento de las diferentes versiones lanzadas por la empresa. Los usuarios dependen de las actualizaciones de los desarrolladores de software y esperan una forma metódica de saber cuándo y qué actualizaciones se publican. Para los proveedores de software, el control de versiones es aún más importante. Realizan un seguimiento de las versiones de software y utilizan la información recopilada para impulsar sus programas de segmentación y monetización. Al separar las actualizaciones principales de las menores, los proveedores pueden destacar, segmentar y cobrar por características específicas. Los proveedores de software también pueden beneficiarse al identificar a los clientes que utilizan la última versión de su software y distinguir a los que aún no se han actualizado. El seguimiento del uso de la versión del cliente reduce el riesgo de que un cliente use software obsoleto y una experiencia de usuario deficiente.

El control de versiones está disponible para el control cambios en el archivo a lo largo del tiempo. Algunos archivos de código o documentación de texto se ejecutan la primera vez que se escriben; Se requieren cambios o reescrituras para corregir errores, cambiar su contenido hasta cierto punto. Los cambios en un documento tienen dos opciones: mantener el historial de cambios o dejar que evolucione sin cambios memoria. El control de versiones es el método predeterminado para mantener esta memoria útil para un mayor desarrollo.

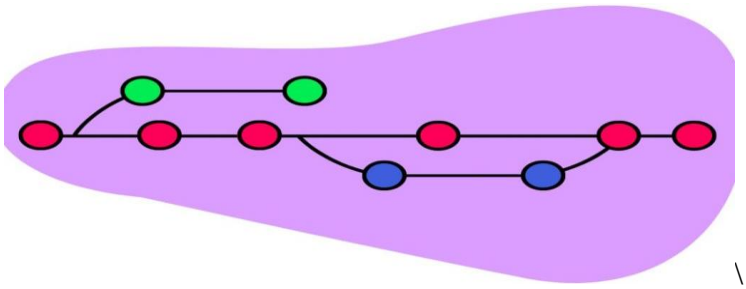
Aunque los sistemas de control de versiones se pueden implementar de forma manual, es muy recomendable utilizar herramientas que faciliten dicha gestión, dando lugar a los denominado Sistemas de Control de Versiones o VCS (Version Control Systems). Estos sistemas ayudan a gestionar las diferentes versiones de cada producto desarrollado, así como las posibles especializaciones (por ejemplo, para clientes específicos). Ejemplos

de tales herramientas son: CVS, Subversion, SourceSafe, ClearCase, Darcs, Bazaar, Plastic SCM, Git, SCCS, Mercurial, Perforce, Fossil SCM, Team Foundation Server.

¿Cuáles son las ventajas de utilizar los sistemas de control de versiones?

1.

Crea ramas y trenzas. Esto es obvio cuando los miembros de su equipo trabajan en conjunto, pero incluso aquellos que trabajan solos pueden beneficiarse de poder gestionar el cambio de forma independiente. La creación de una "sucursal" en una herramienta VCS permite que múltiples flujos de trabajo sean independientes entre sí, al tiempo que brinda la capacidad de fusionarse nuevamente en ese trabajo, lo que permite a los desarrolladores verificar que los cambios de cada rama no hayan cambiado. Estarán en conflicto. Muchos equipos de software utilizan la práctica de crear ramas para cada función o cada versión, o ambas. Los equipos pueden elegir entre muchos flujos de trabajo diferentes cuando deciden cómo usar la bifurcación y la fusión en VCS.



\

Con esto lo que se busca es que se pueda trabajar de forma paralela en distintas versiones a la vez para así brindar mejoras a la aplicación sin necesidad de estar deteniendo el funcionamiento de la misma y no interrumpir a otros miembros del equipo que se encuentren trabajando también en el sistema o aplicación.

2.

Historial completo de cambios a largo plazo para todos los archivos. Esto significa todos los cambios que muchas personas han hecho a lo largo de los años. Los cambios incluyen la creación y eliminación de archivos, así como cambios en su contenido. Diferentes herramientas de VCS manejan el cambio de nombre y el movimiento de archivos de manera diferente. Este historial también debe incluir el autor, la fecha y la descripción escrita de cada cambio. Con un historial completo, puede volver a versiones anteriores para ayudar a analizar la causa de los errores, lo cual es esencial cuando se solucionan problemas de versiones de software anteriores. Si está desarrollando software activamente, casi cualquier cosa puede considerarse una "versión antigua" de software.

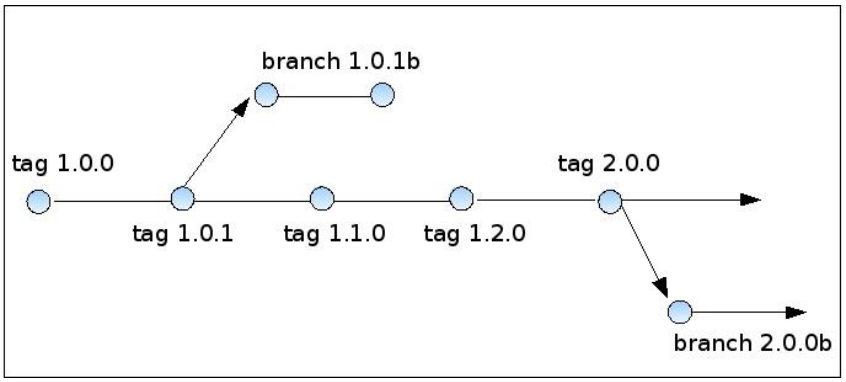
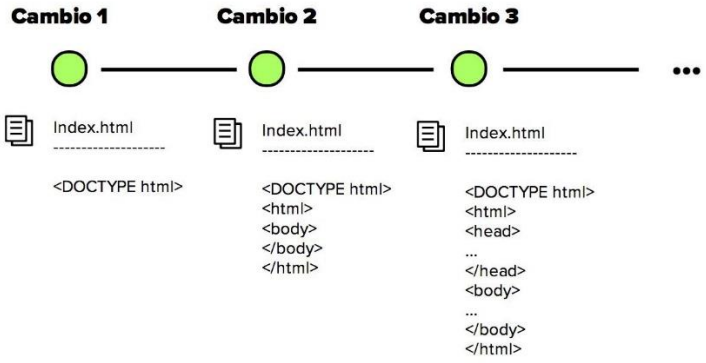


Ilustración 1 Ejemplos de versiones de un sistema.

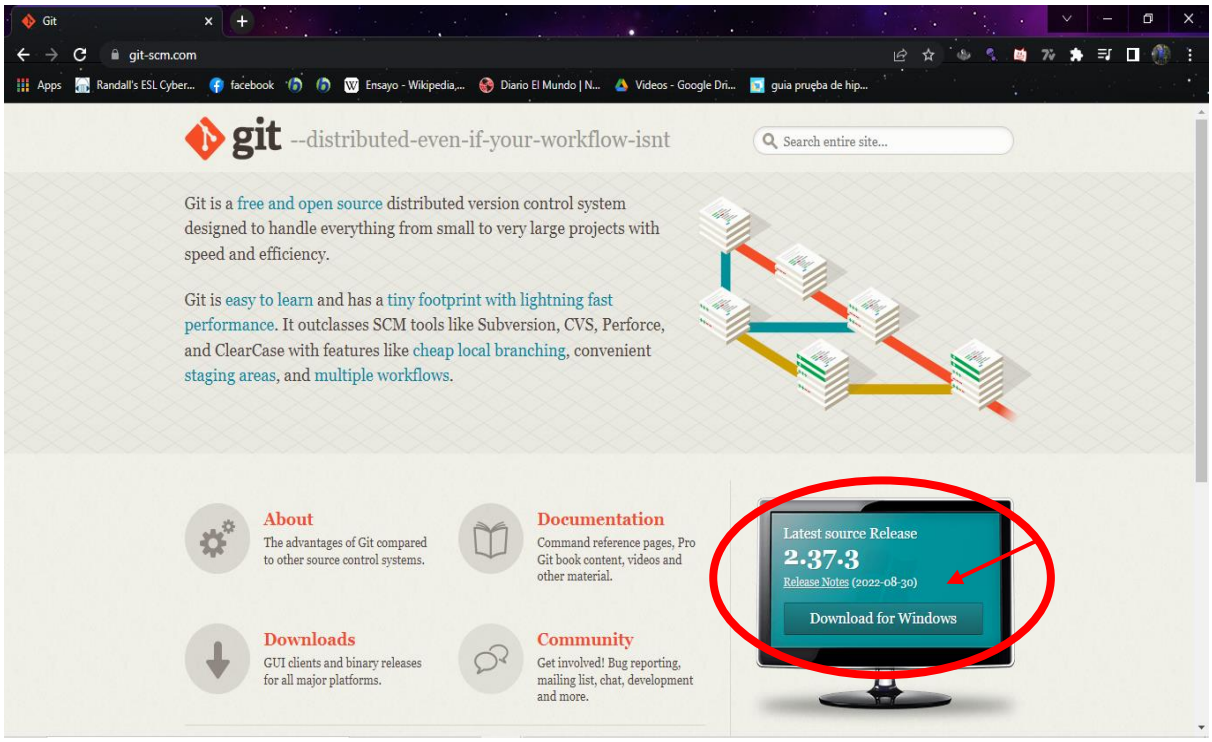
3.

Trazabilidad, ser capaz de realizar un seguimiento de todos los cambios de software y conectarlo a un software de gestión de proyectos y seguimiento de errores como Jira, y poder comentar cada cambio con un mensaje que describa el propósito y los objetivos del cambio, solo ayuda a abordar la causa raíz. análisis y recopilación de información. Un historial de código anotado, en la punta de los dedos de la lectura del código, tratando de entender lo que hace y por qué está diseñado de la manera que está, permite a los desarrolladores realizar cambios correctos y armoniosos de acuerdo con el código a largo plazo esperado, el diseño de la terminología del sistema. Esto es especialmente importante para la gestión eficaz del código heredado y es esencial para que los desarrolladores calculen con precisión el trabajo futuro.

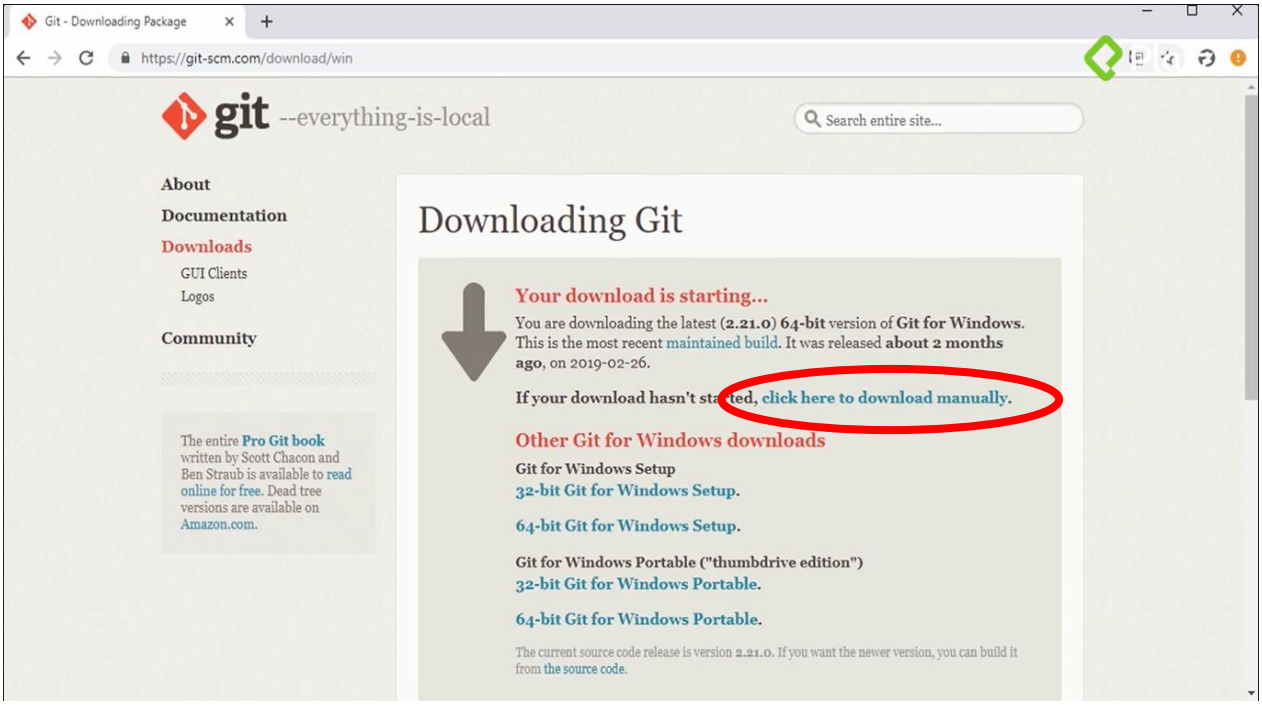


PROCESO DE INSTALACION DE GIT EN SISTEMA OPERATIVO WINDOWS:

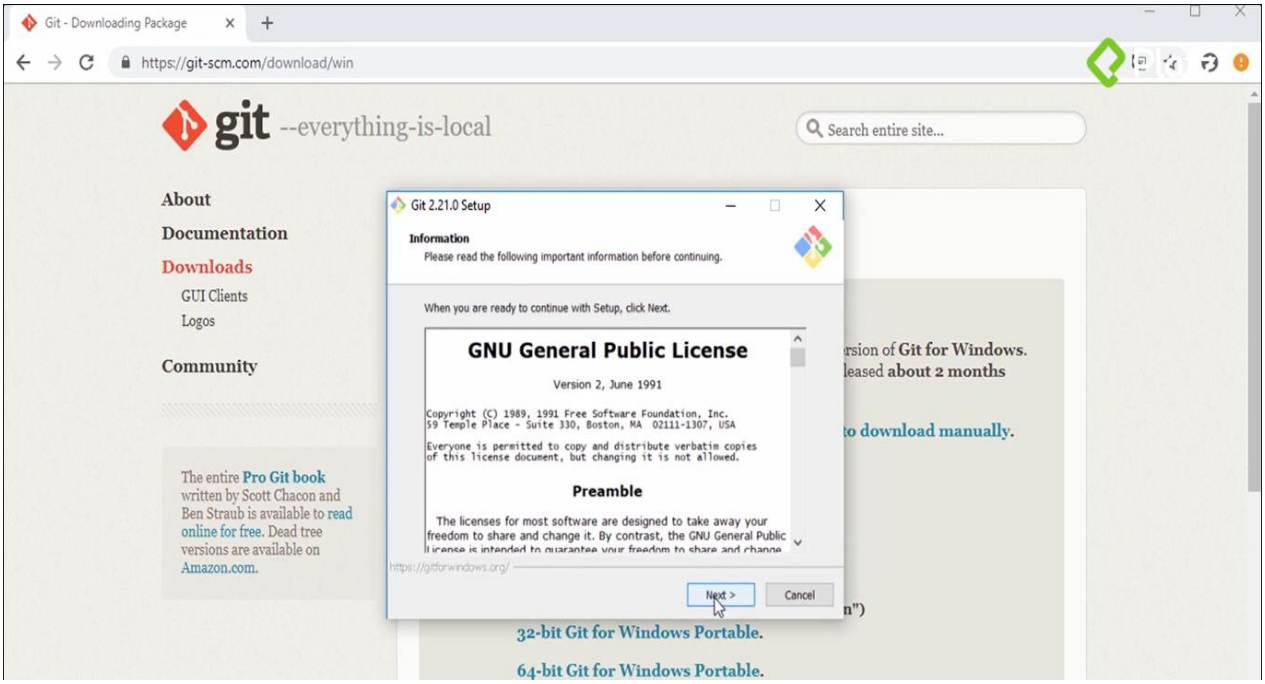
1. **Primero que todo para instalar Git y GitBash en un sistema operativo Windows, lo primero que se debe hacer es abrir un navegador (cualquiera sea de su preferencia) como por ejemplo Chrome, y buscar el siguiente URL https://git-scm.com/ con el cual se le mostrara esta ventana, y debe darle clic a lo que se le indica en la imagen con una flecha y un círculo:**

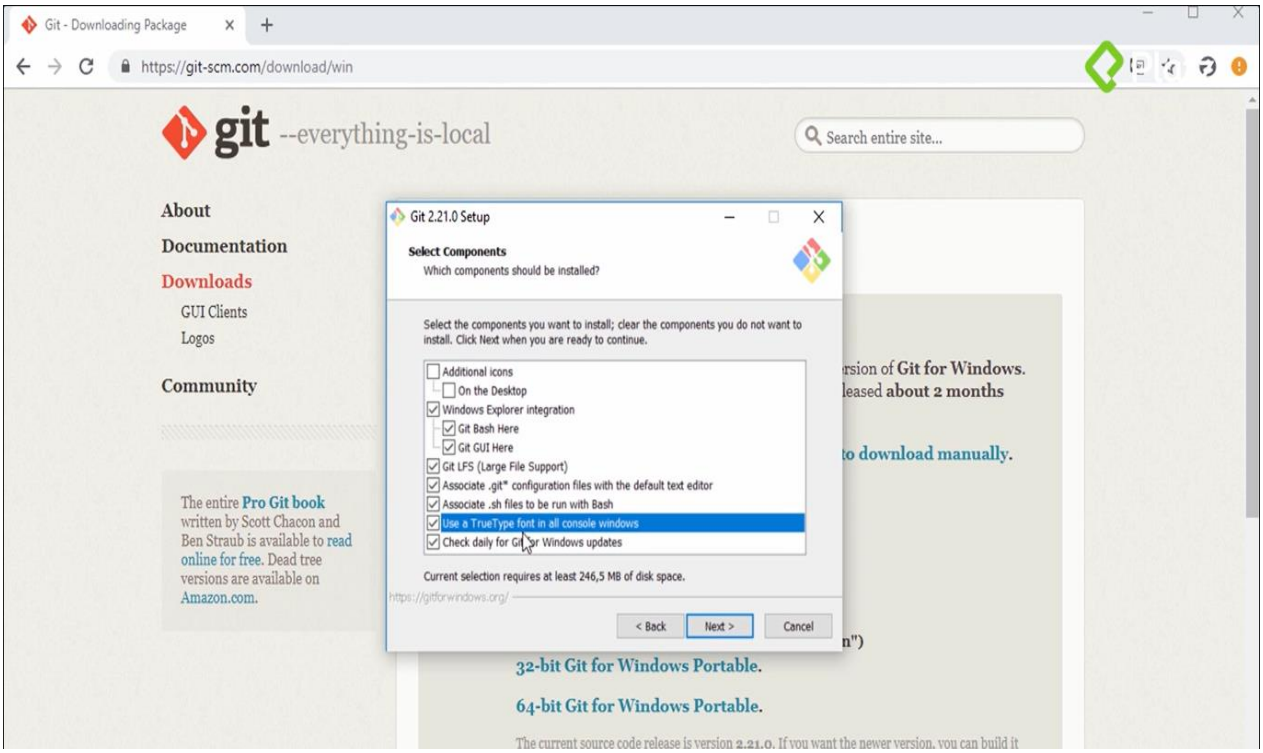


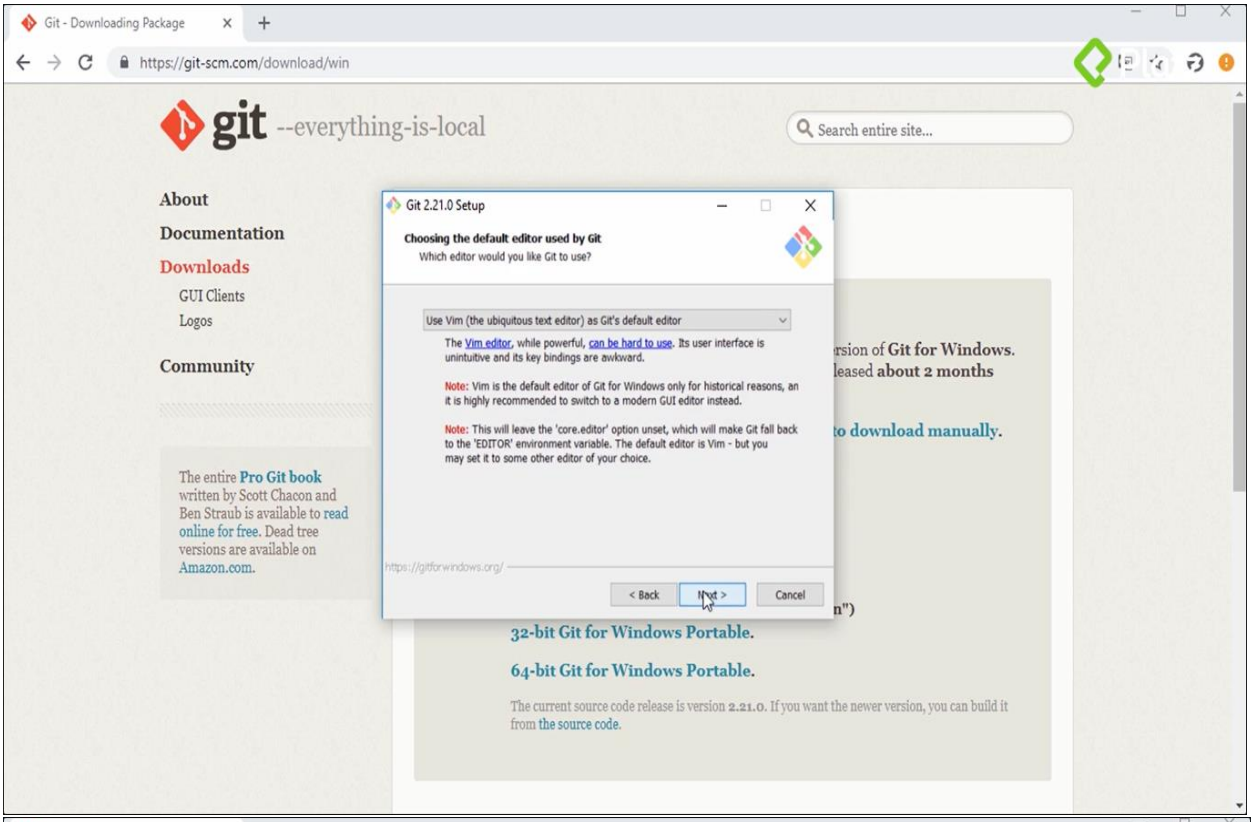
2. Luego de acceder a ese link se le mostrara la siguiente ventana, con la cual debe dar clic en lo que se le indica con el círculo rojo

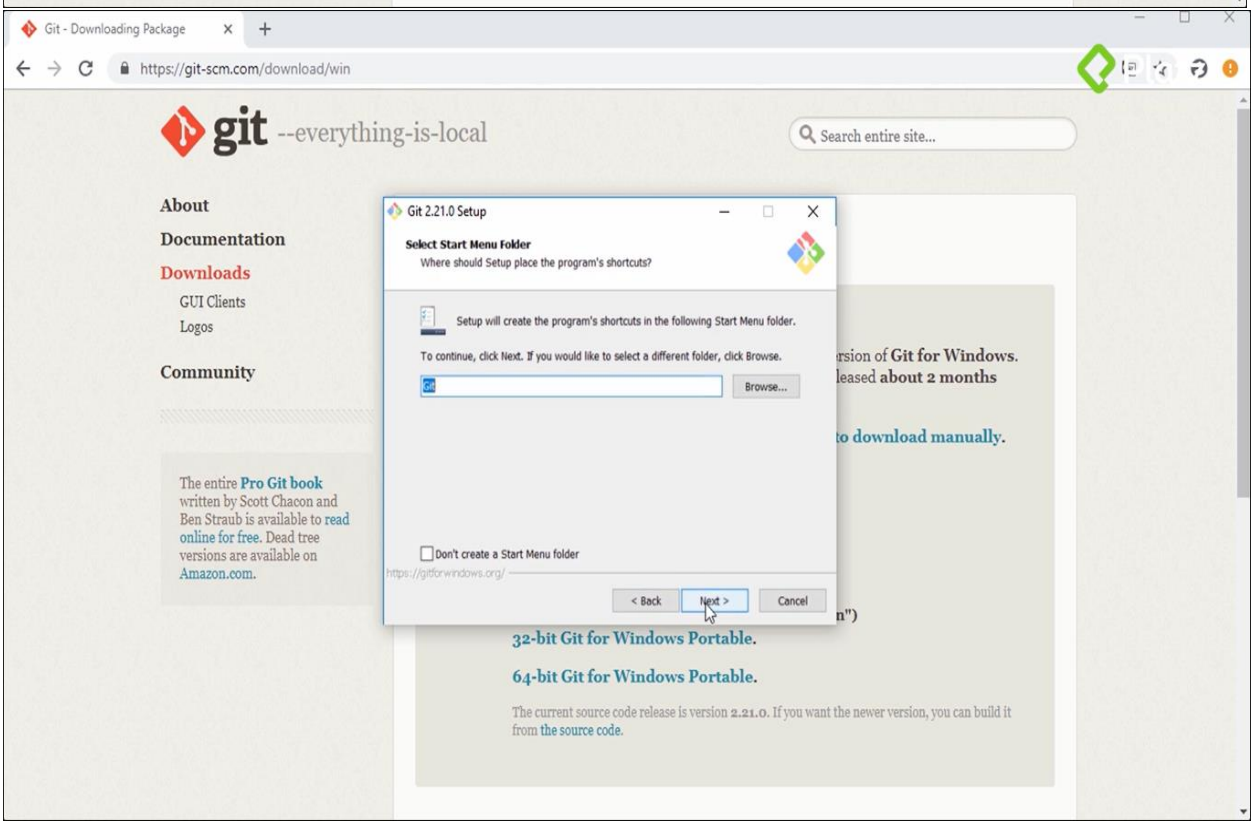


3. Paso siguiente a esto, se le mostraran varias ventanas, a las cuales tiene que dar únicamente "Next" y continuar hasta que se instale el sistema en su computador.

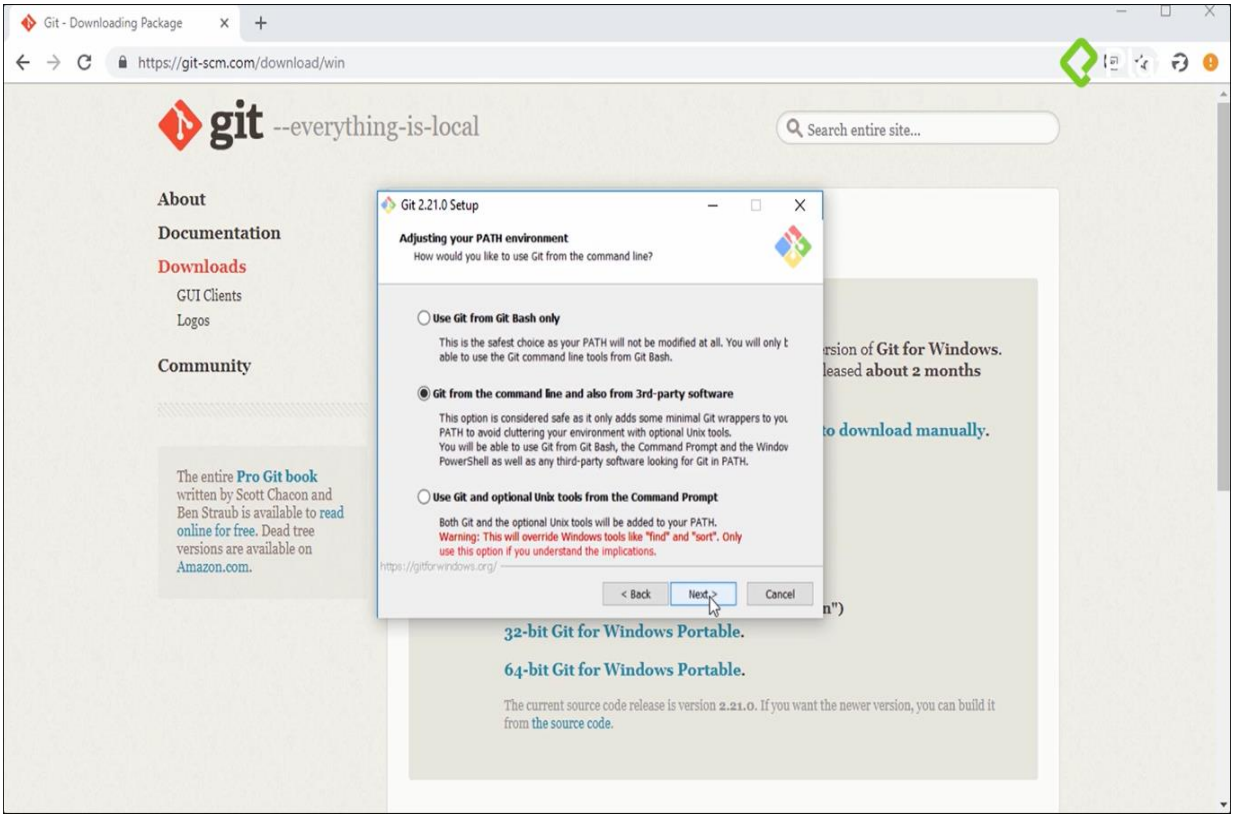


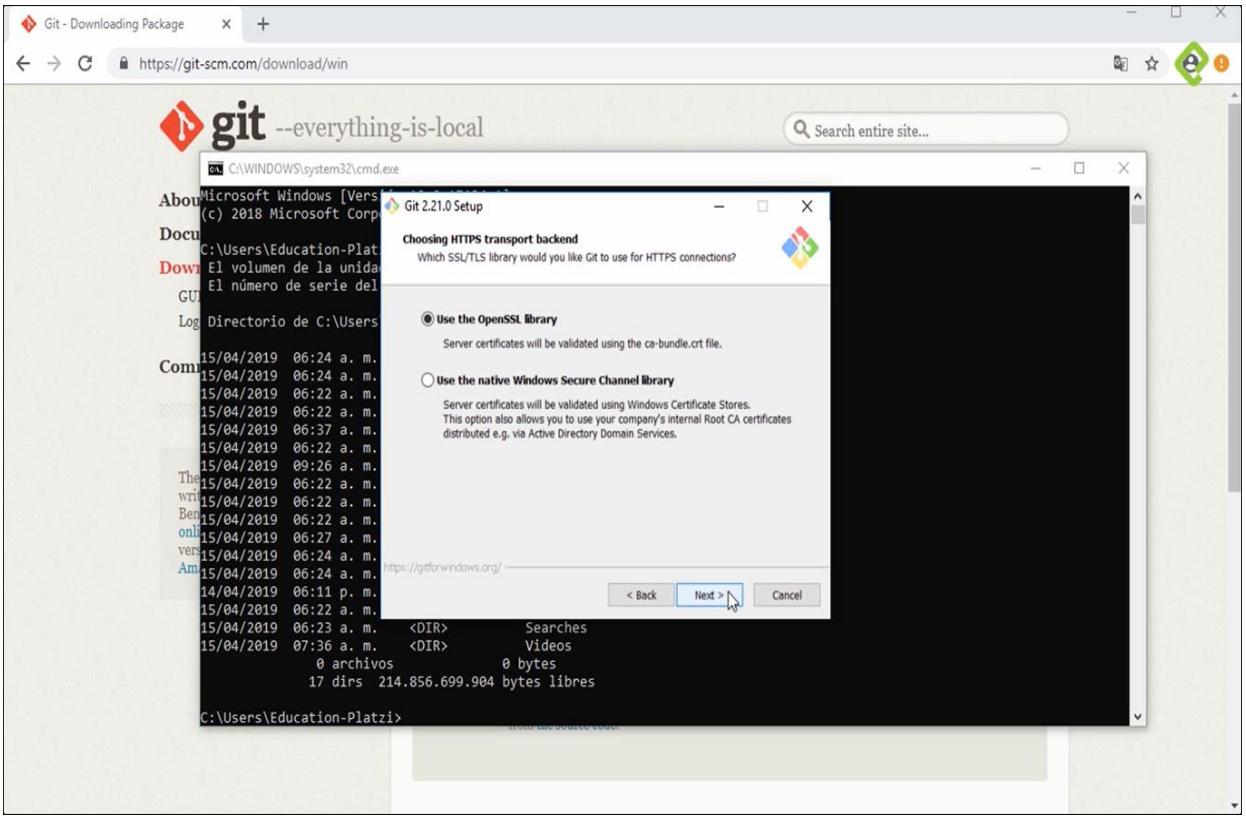

Git - Downloading Package
https://git-scm.com/download/win
git --everything-is-local
Search entire site...
About
Documentation
Downloads
GUI Clients
Logos
Community
The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.
Git 2.21.0 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Git Bash Here
Git GUI Here
Git LFS (Large File Support)
Associate .git* configuration files with the default text editor
Associate .sh files to be run with Bash
Use a TrueType font in all console windows
Check daily for Git for Windows updates
Current selection requires at least 246,5 MB of disk space.
https://gitforwindows.org/
< Back
Next >
Cancel
32-bit Git for Windows Portable.
64-bit Git for Windows Portable.


Git - Downloading Package
https://git-scm.com/download/win
git --everything-is-local
Search entire site...
About
Documentation
Downloads
GUI Clients
Logos
Community
The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.
Git 2.21.0 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Vim (the ubiquitous text editor) as Git's default editor
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.
Note: Vim is the default editor of Git for Windows only for historical reasons, an it is highly recommended to switch to a modern GUI editor instead.
Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.
https://gitforwindows.org/
< Back
Next >
Cancel
rsion of Git for Windows.
leased about 2 months
to download manually.
n")
32-bit Git for Windows Portable.
64-bit Git for Windows Portable.
The current source code release is version 2.21.0. If you want the newer version, you can build it from the source code.

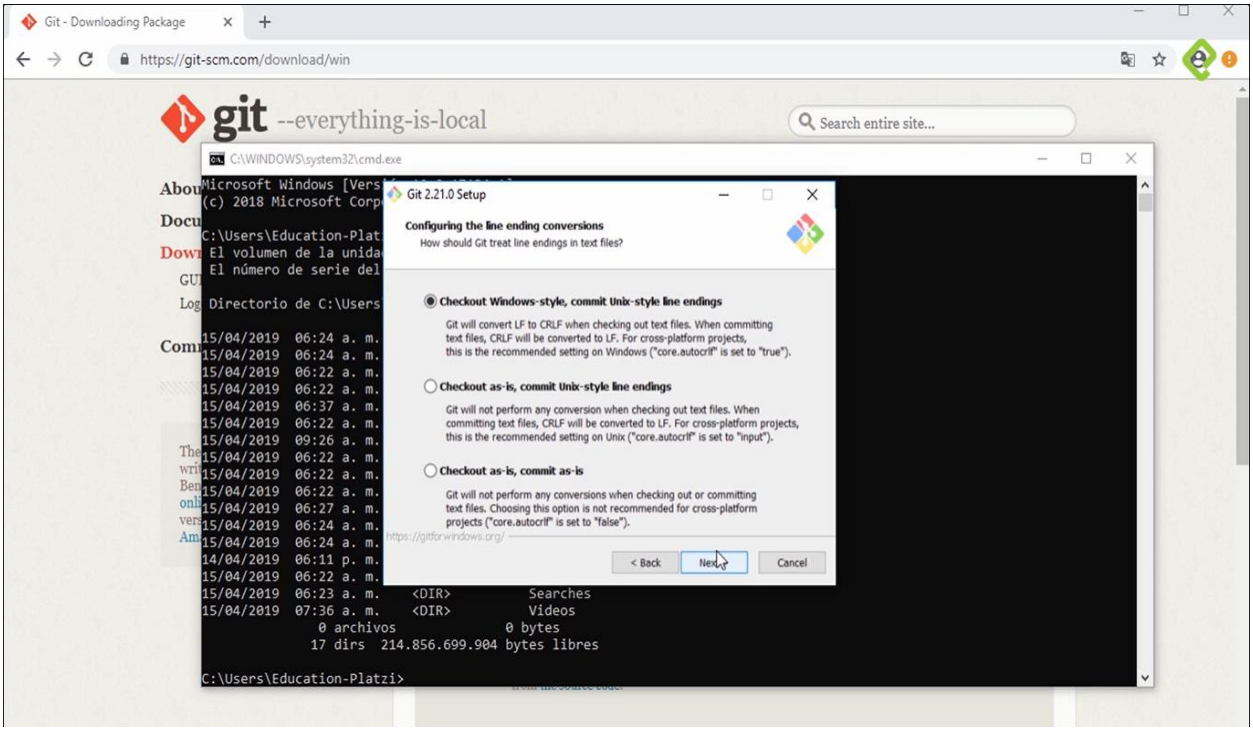

Git - Downloading Package
https://git-scm.com/download/win
git --everything-is-local
Search entire site...
About
Documentation
Downloads
GUI Clients
Logos
Community
The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.
Git 2.21.0 Setup
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
Git
Browse...
Don't create a Start Menu folder
https://gitforwindows.org/
< Back
Next >
Cancel
rsion of Git for Windows.
leased about 2 months
to download manually.
n")
32-bit Git for Windows Portable.
64-bit Git for Windows Portable.
The current source code release is version 2.21.0. If you want the newer version, you can build it from the source code.

4. En esta parte debe de seleccionar la opción de en medio "Git from the command line and also from 3rd parrty Software". Con esto permitimos que Git pueda utilizarse desde la línea de comandos que ya es nativa en Windows y además el software como Gitbash entre otros. Y seguimos dando "Next".

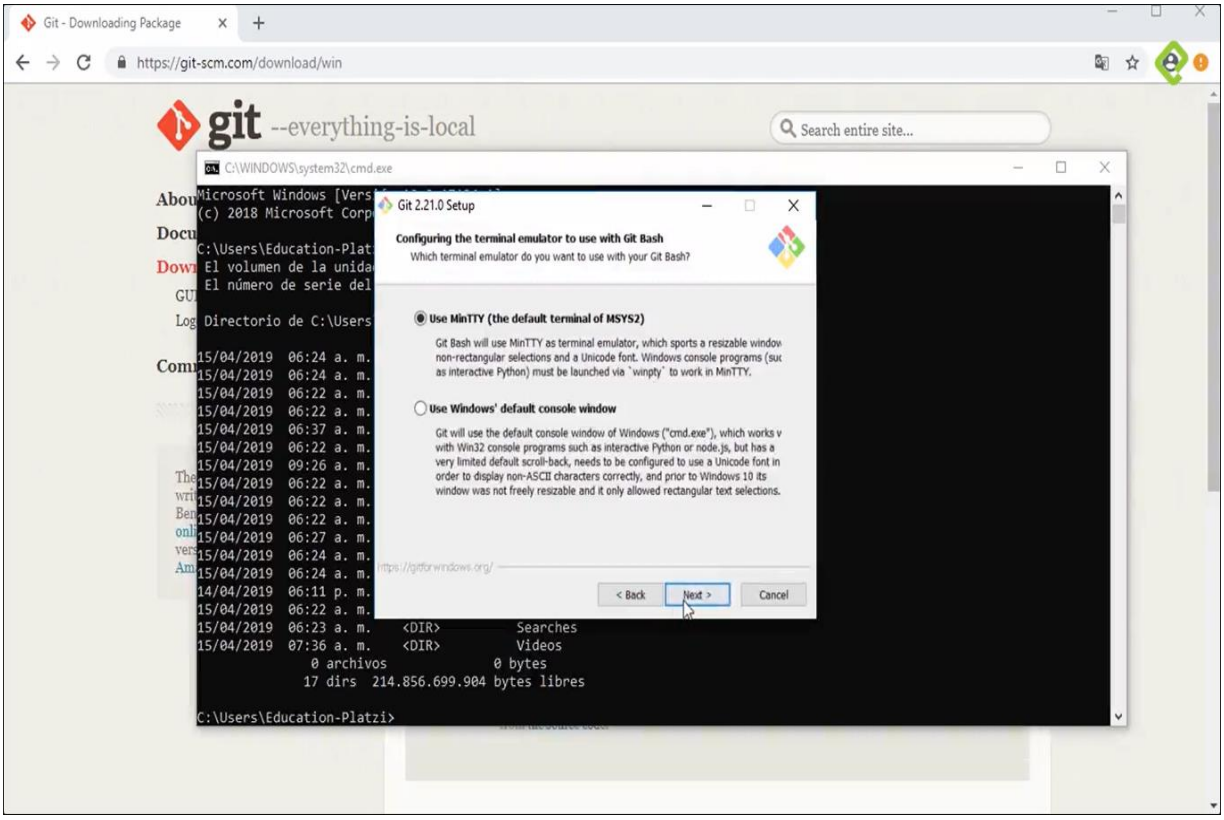




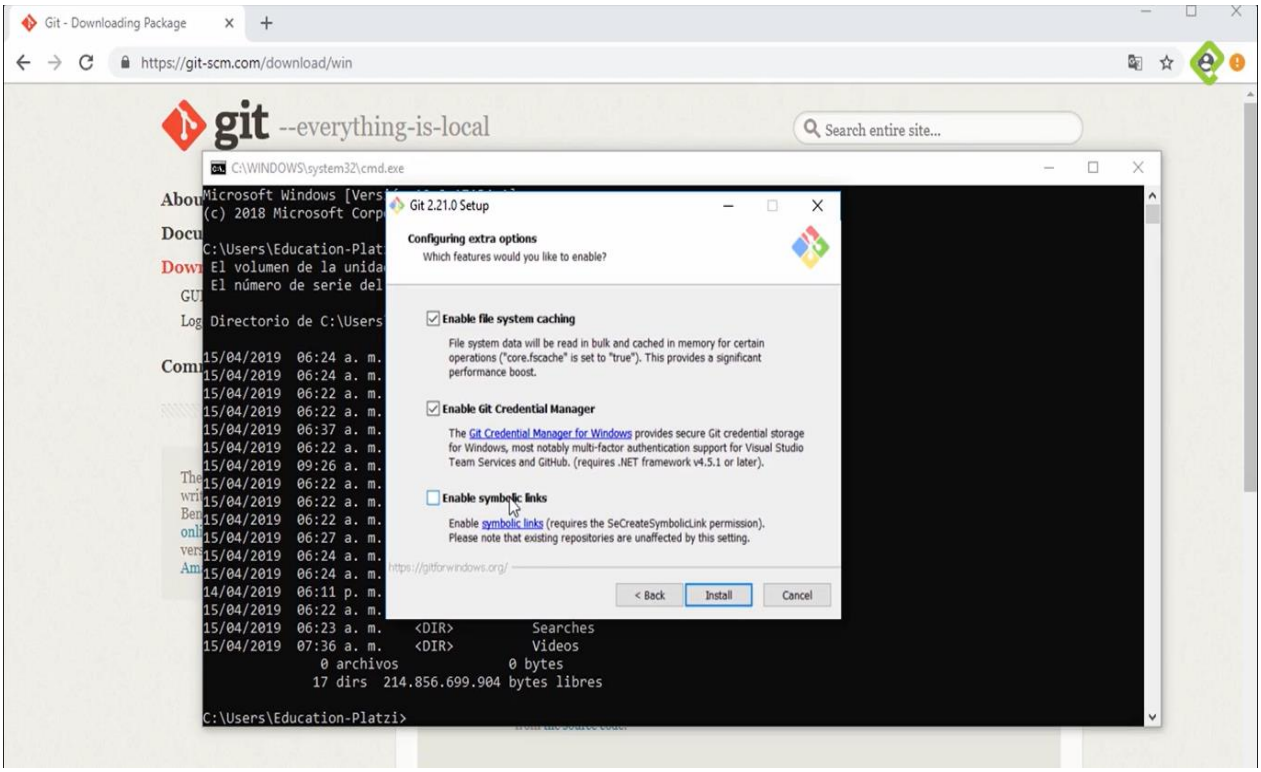
5.Aqui escoger siempre la opcion#1, la de "Chechout Windows-Style, commit unix-style line endings", y seguir dando "Next".

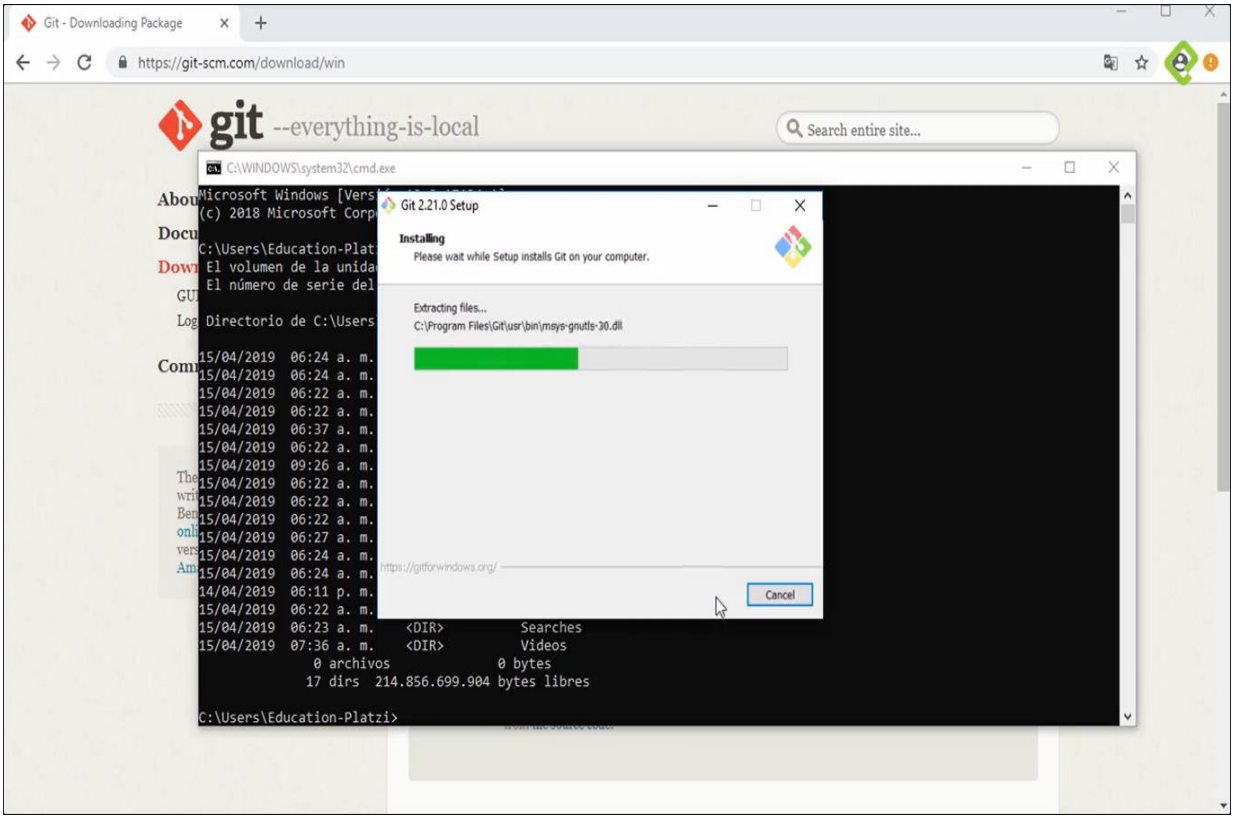


6.Luego escoger igual la primera opcion "Use MainTTY", y dar Next

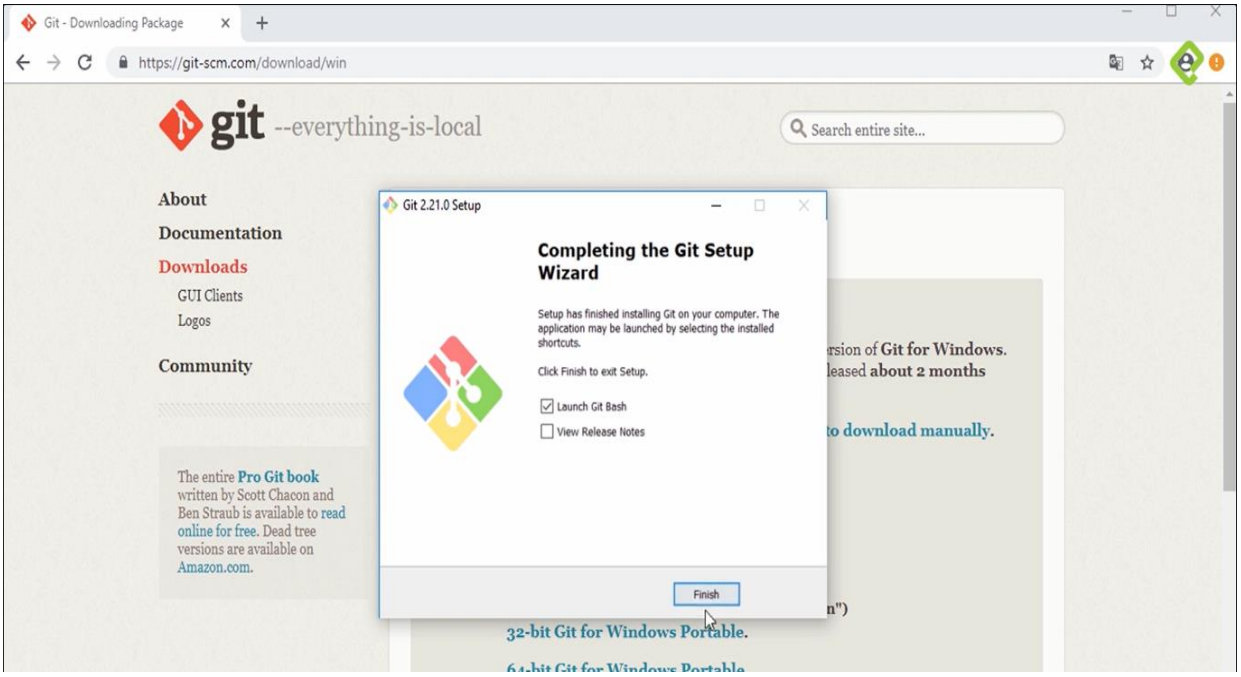


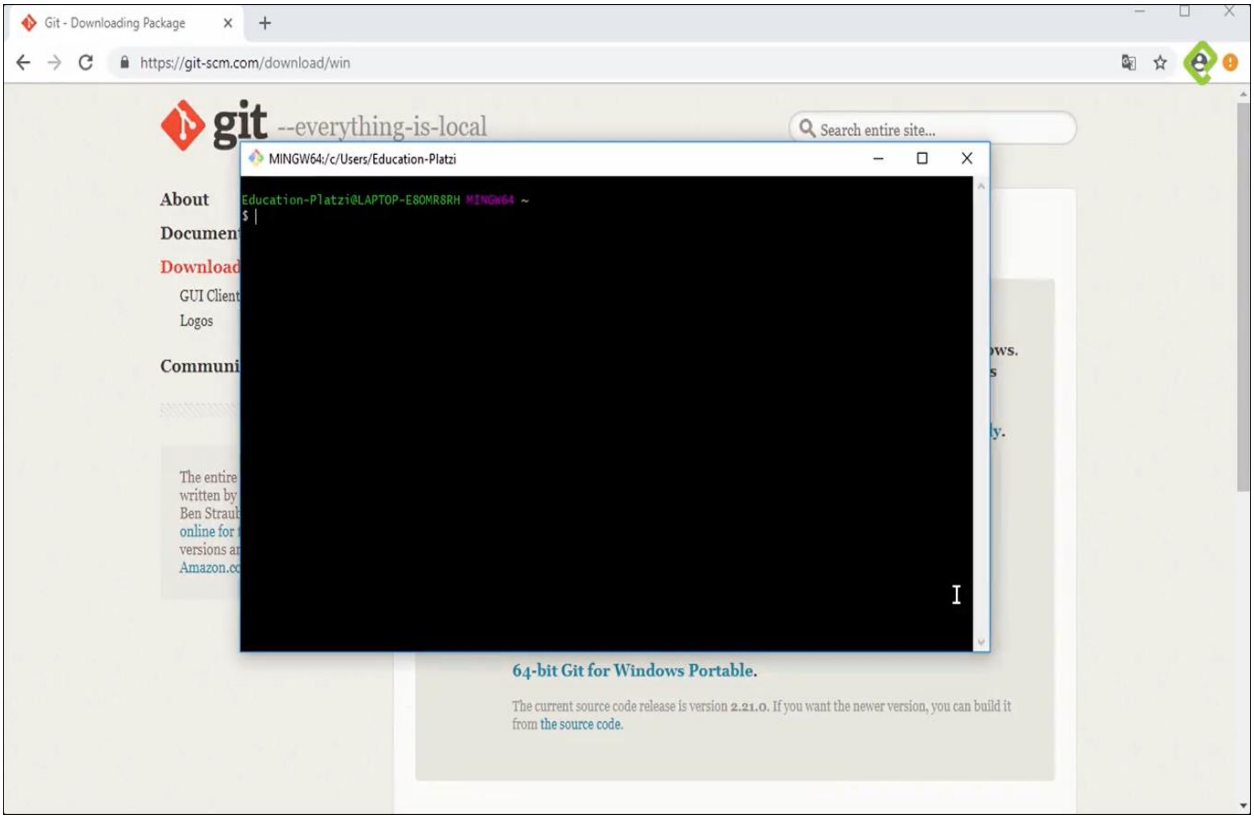
7. Ya después de haber seguido todo correctamente, solo queda dar clic en "Install" y listo.





8. Como paso final, seleccionar la opcion de “Launch Git Bash” y en pantalla aparecerá una ventana con fondo negro, y esto es lo que conocemos como Git.





9. Listo, ya tiene instalado Git en su sistema operativo de Windows, con el que puede realizar diversos comandos para guardarlos en un repositorio ya sea interno o en la nube.