Nombre de la práctica				No.	
Asignatura:		Carrera:		Duración de la práctica (Hrs)	

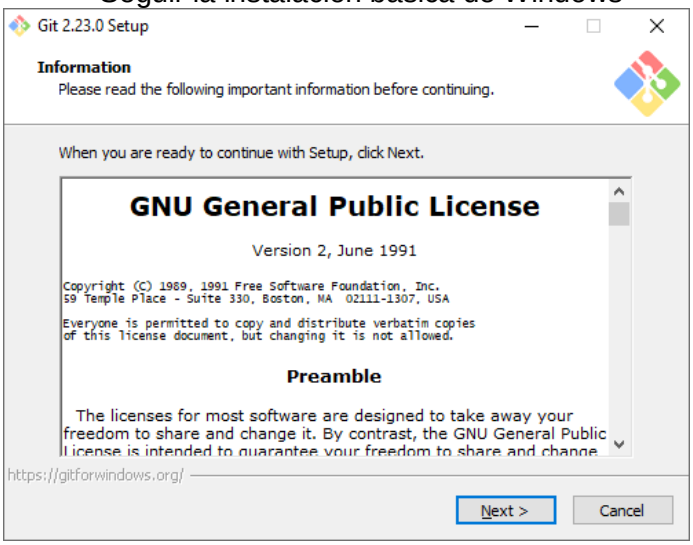
I. Competencia(s) específica(s):

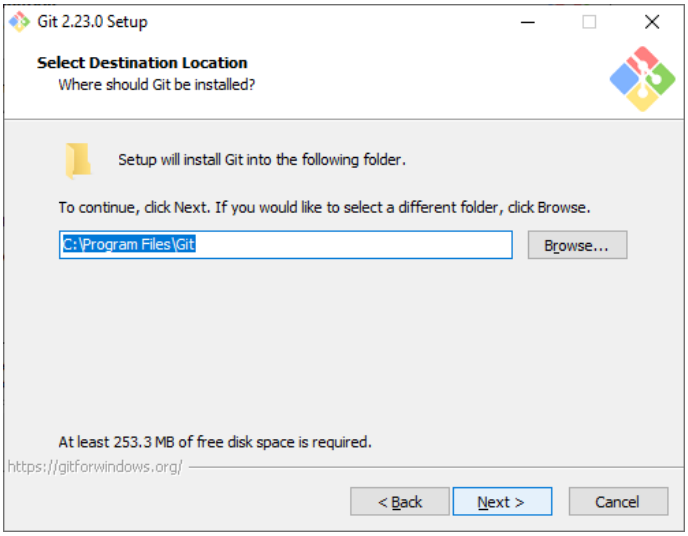
II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

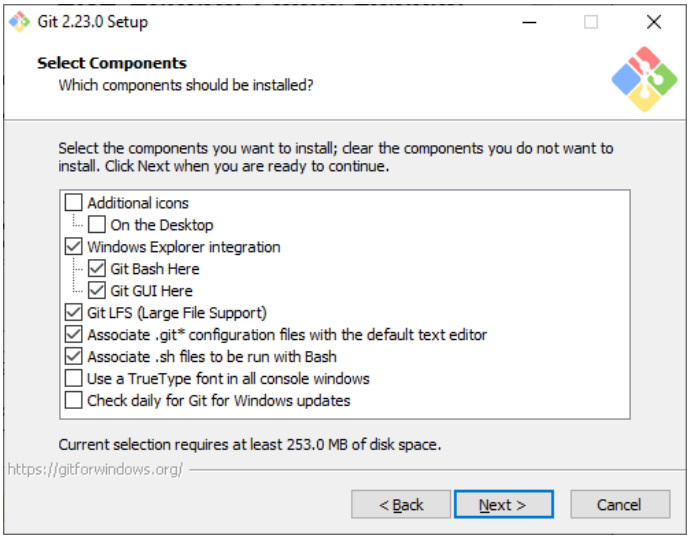
III. Material empleado:

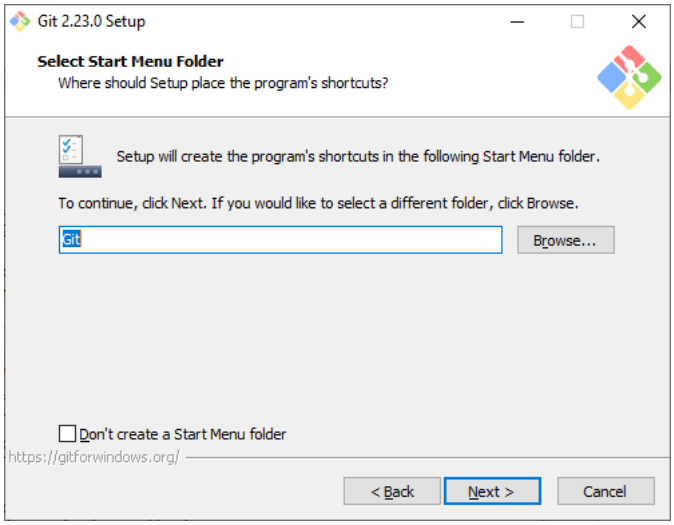
IV. Desarrollo de la práctica:

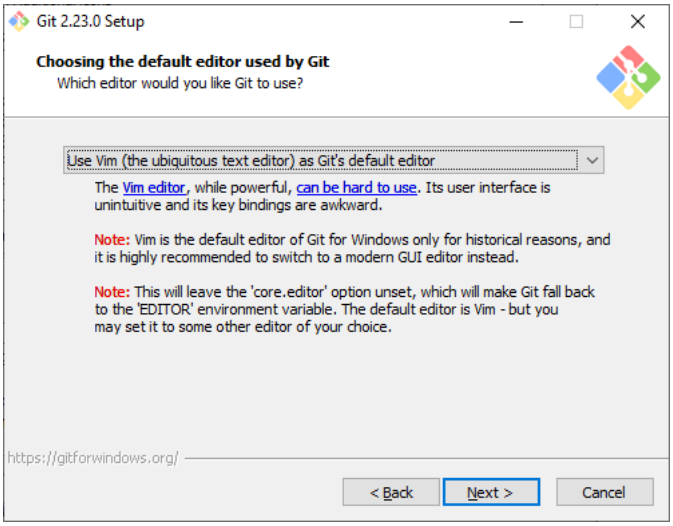
Seguir la instalación básica de Windows

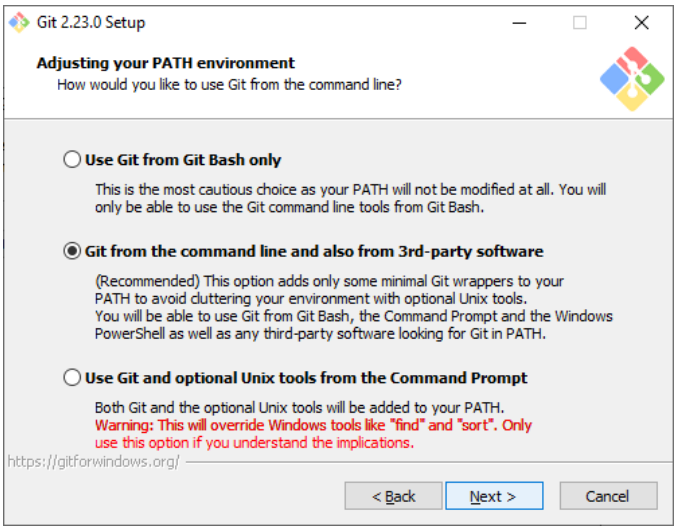


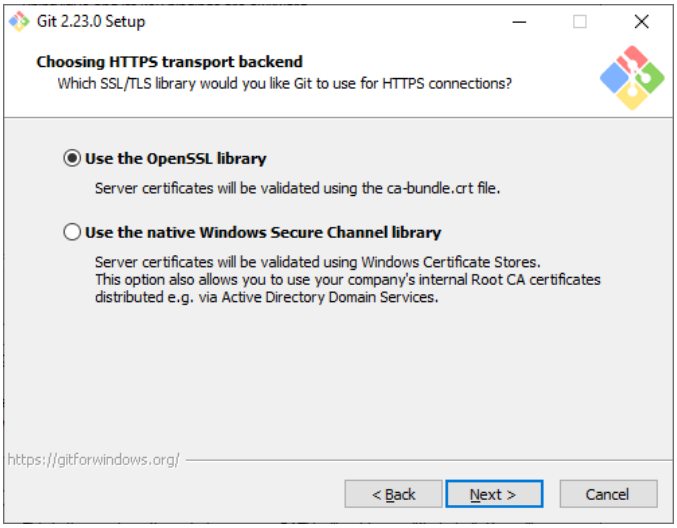


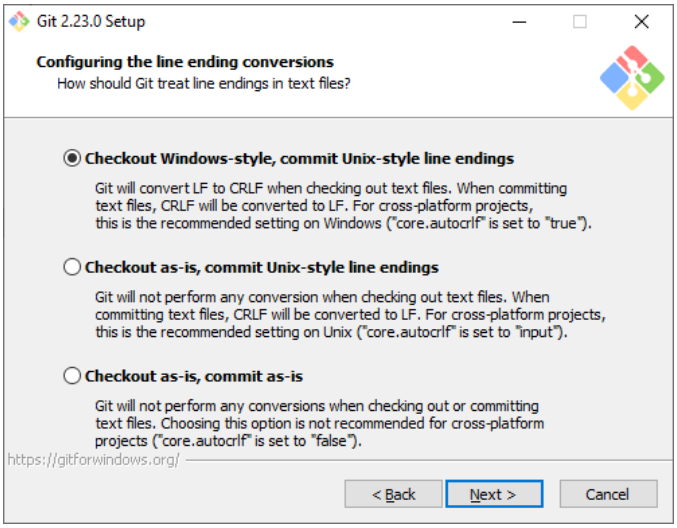


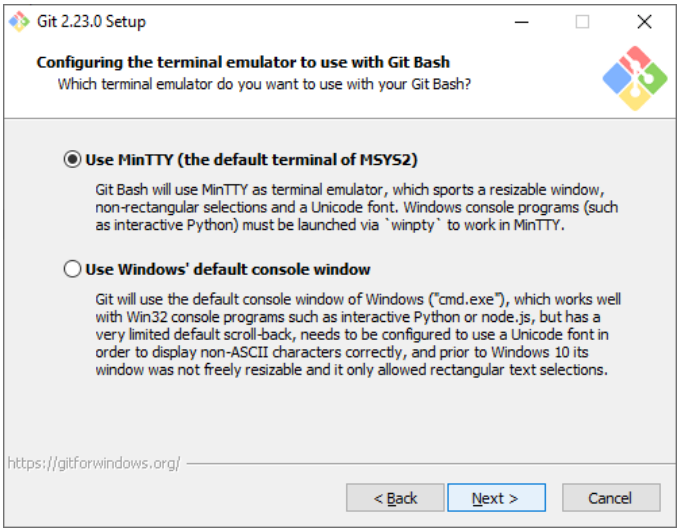


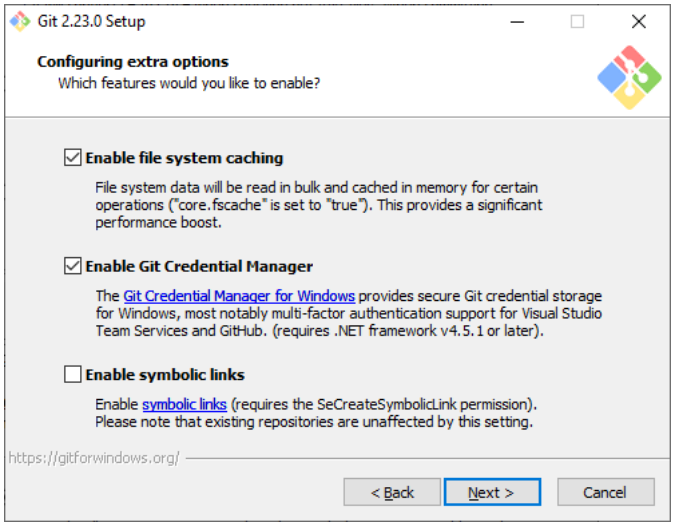

Git 2.23.0 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Vim (the ubiquitous text editor) as Git's default editor
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.
Note: Vim is the default editor of Git for Windows only for historical reasons, and 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

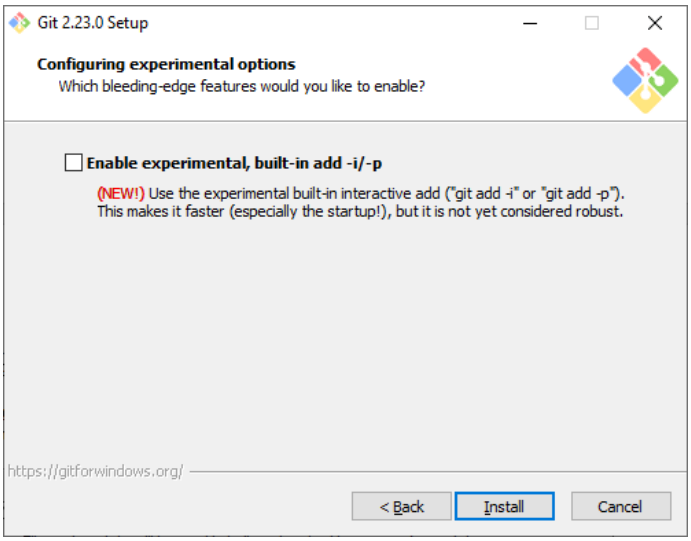

Git 2.23.0 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Git from the command line and also from 3rd-party software
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.
Use Git and optional Unix tools from the Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.
https://gitforwindows.org/
< Back
Next >
Cancel

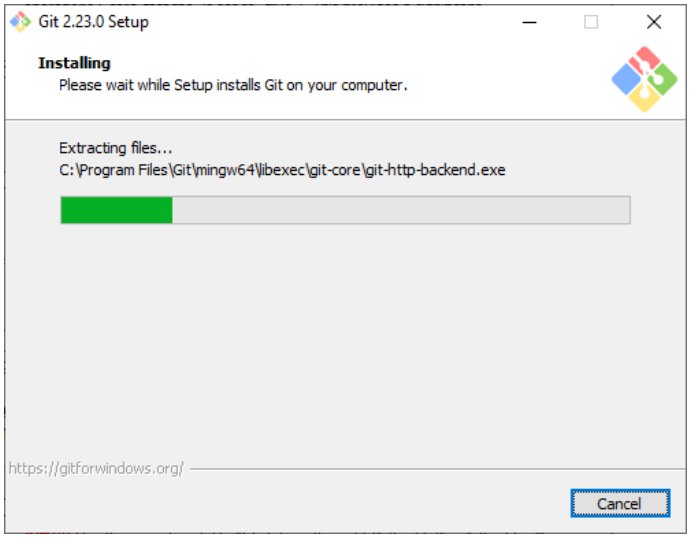

Git 2.23.0 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using 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

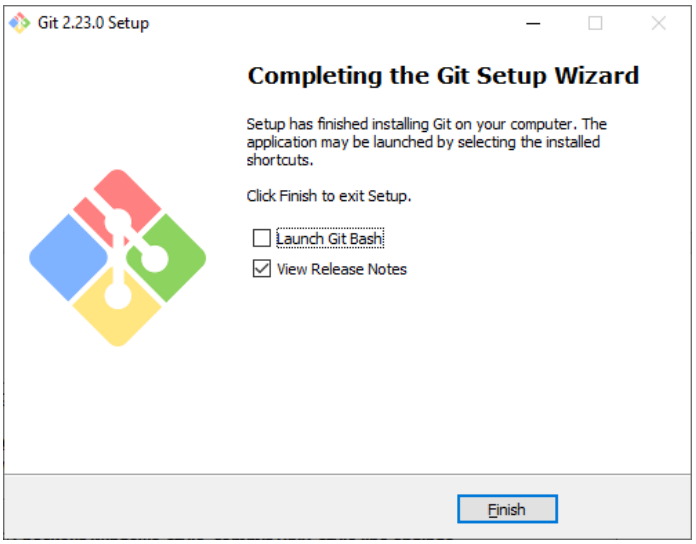

Git 2.23.0 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, 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 2.23.0 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 2.23.0 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 Git Credential Manager
The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).
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



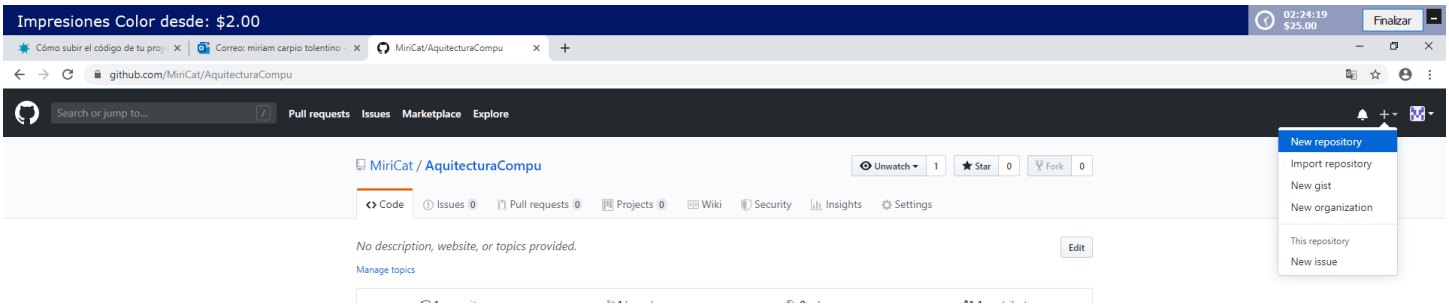




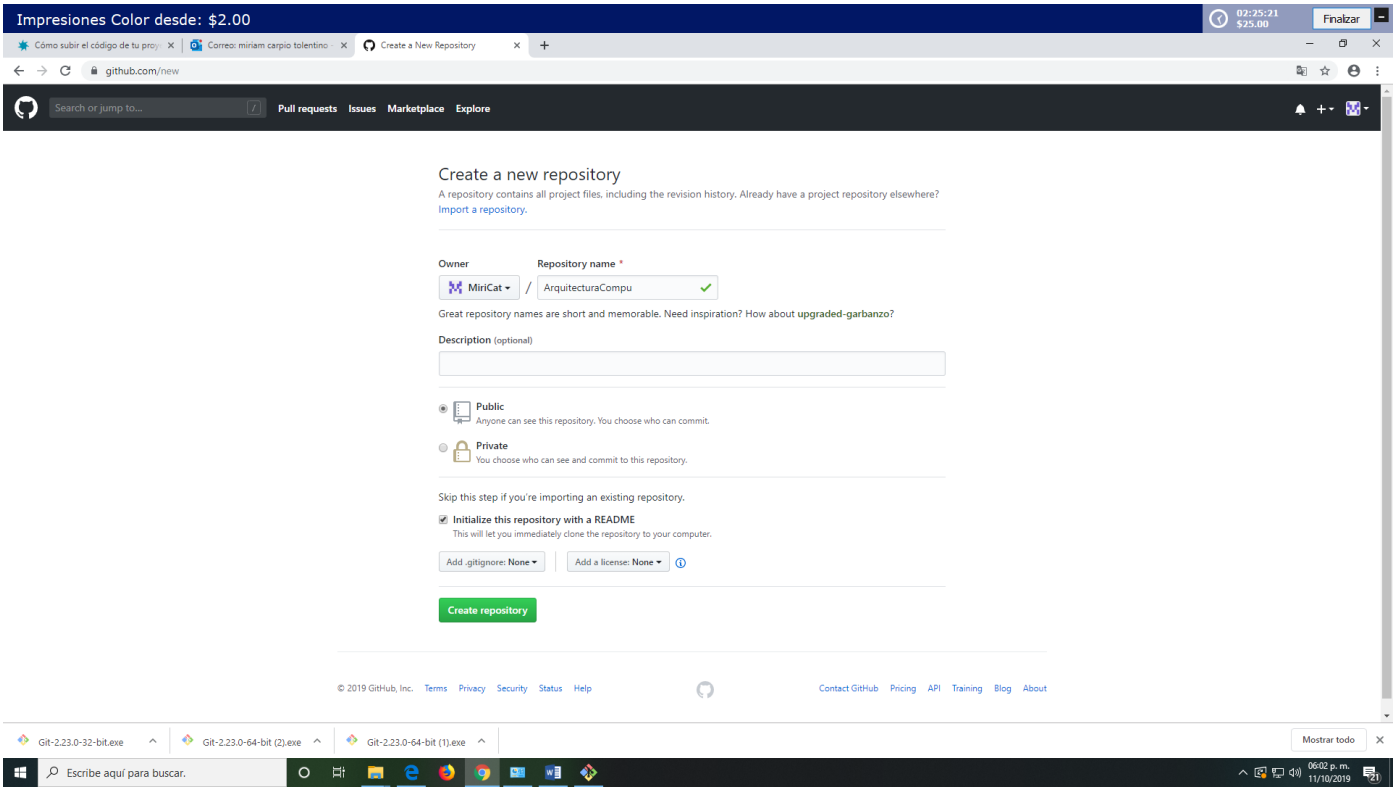
Checamos loa versión

Entramos a Github

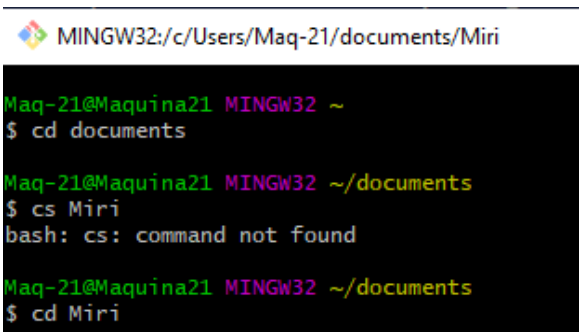
Creamos un reposiroeio nuevo



Colocamos por nombre ArquitecturaCompu



En la consola de Git bash entramos a la ruta donde esta nuestro documento, en mi casa esta en Documentos/Miri



Colocamos el comando **git initi** /comando para crear un repositorio local

```
Maq-21@Maquina21 MINGW32 ~/documents/Miri
$ git init
Initialized empty Git repository in C:/Users/Maq-21/documents/Miri/.git/
```

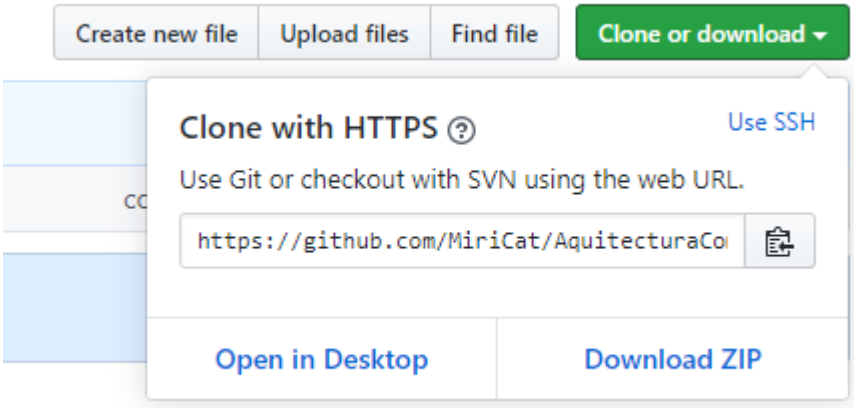
Añadir todos los archivos del proyecto con **git add .**

```
Maq-21@Maquina21 MINGW32 ~/documents/Miri (master)
$ git add .
```

Hacer un commit por cada cambio

```
Maq-21@Maquina21 MINGW32 ~/documents/Miri (master)
$ git commit -m 'commit22'
[master (root-commit) e795f70] commit22
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 3051_Carpio,Tolentino-Miriam_Tarea.pdf
```

Subir los cambios al repositorio pero para esto debemos ir a github y copiar la ruta



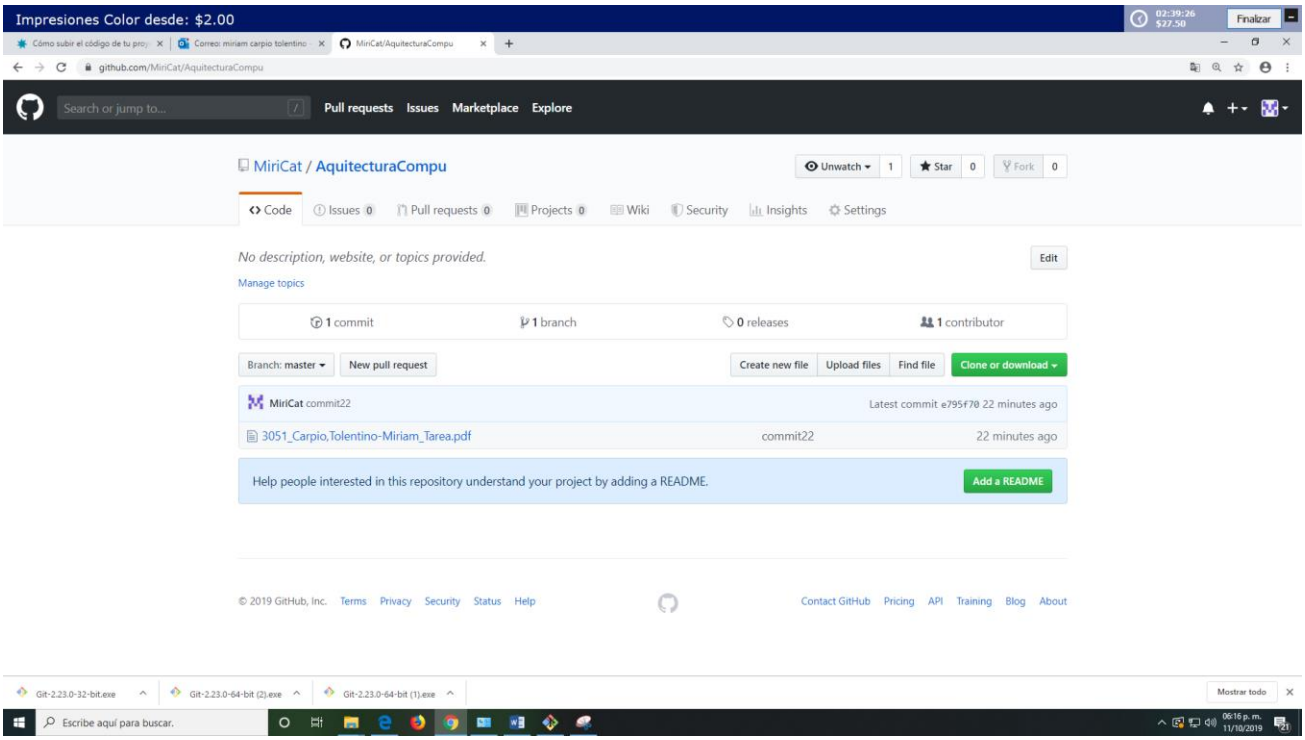
Colocamos **git remote add origin ruta**

```
Maq-21@Maquina21 MINGW32 ~/documents/Miri (master)
$ git remote add origin https://github.com/MiriCat/AquitecturaCompu.git
```

Subir los cabios al repositorio con push

```
Maq-21@Maquina21 MINGW32 ~/documents/Miri (master)
$ git push --force origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 406.42 KiB | 23.91 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/MiriCat/AquitecturaCompu.git
 + 2142a33...e795f70 master -> master (forced update)
```

Ir a github y refrescar la pagina



V. Conclusiones:

Para subir archivos a git use git bash que es para el sistema operativo Windows. Usa comandos de Linux.

Esta herramienta nos sirve para trabajar en equipo desde diferentes partes, solo con la ruta del proyecto, es usado atreves de repositorios