

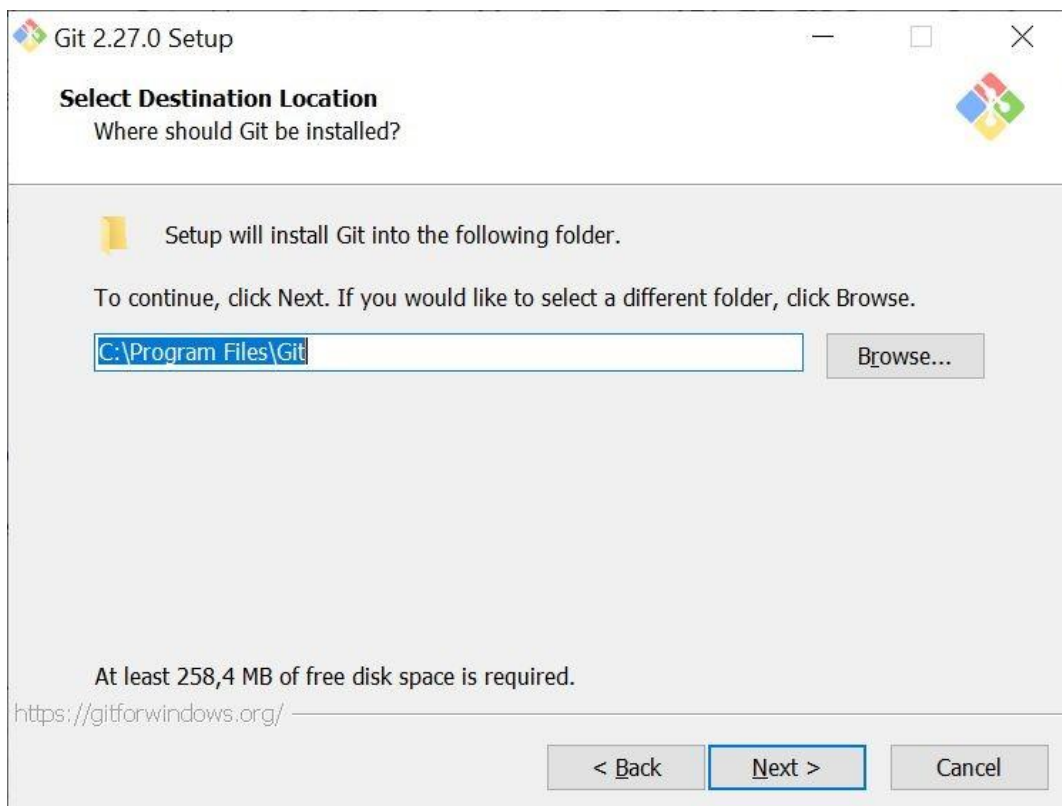
Taller de Git y GitHub

Para la realización de este taller se requiere las siguiente herramientas:

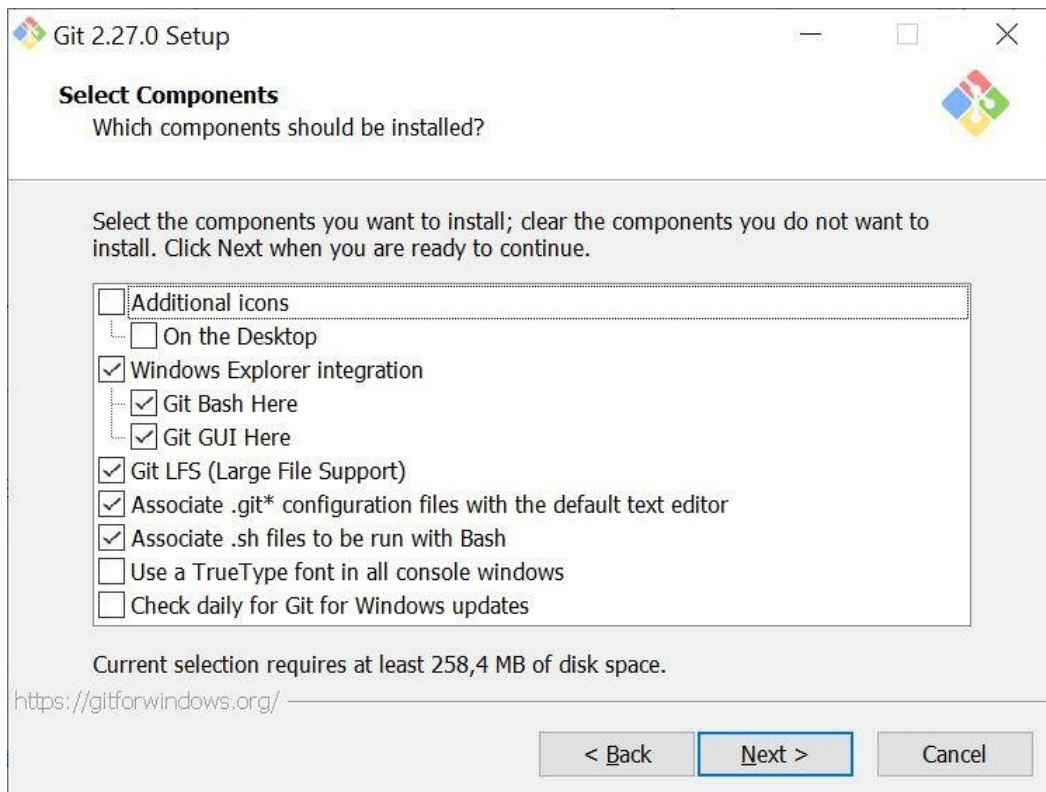
- Sistema Operativo Microsoft Windows
- Git for windows <https://git-scm.com/download/win>
- Sourcetree <https://www.sourcetreeapp.com/>

Instalación de Git

1. Ejecutar el instalador **Git-2.27.0-64-bit.exe**.
2. Aceptar la licencia.
3. Seleccionar la carpeta de instalación.

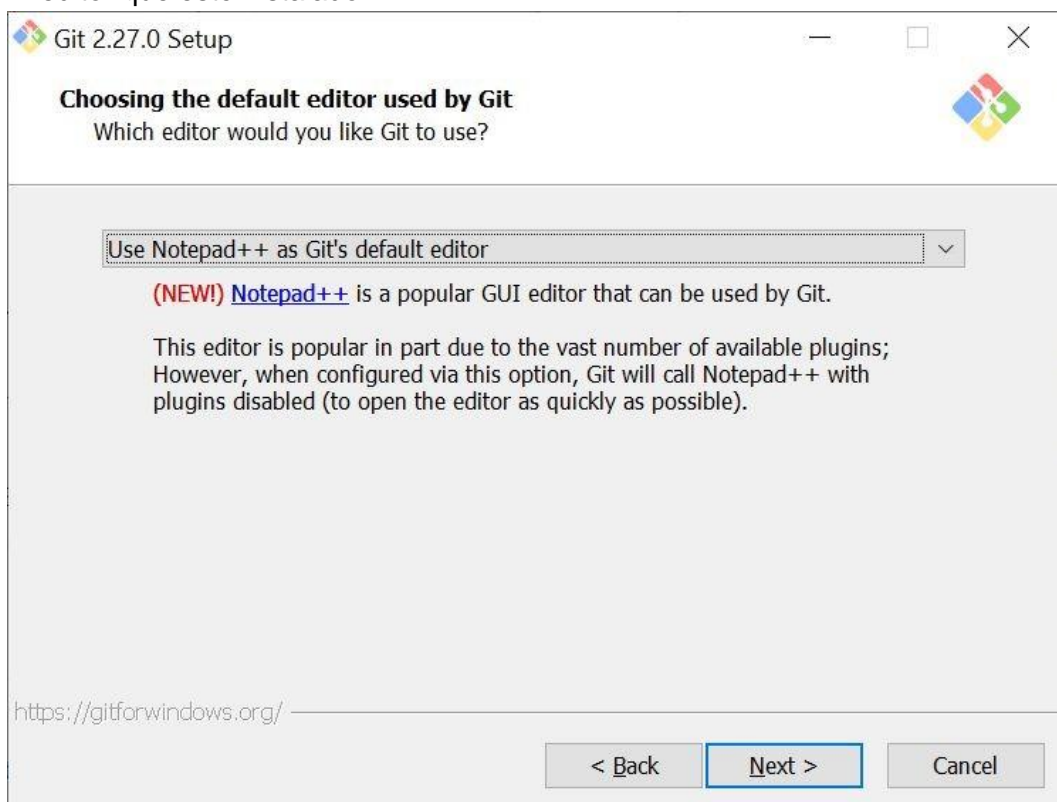


4. Seleccionar componentes adicionales. No hacer nada.

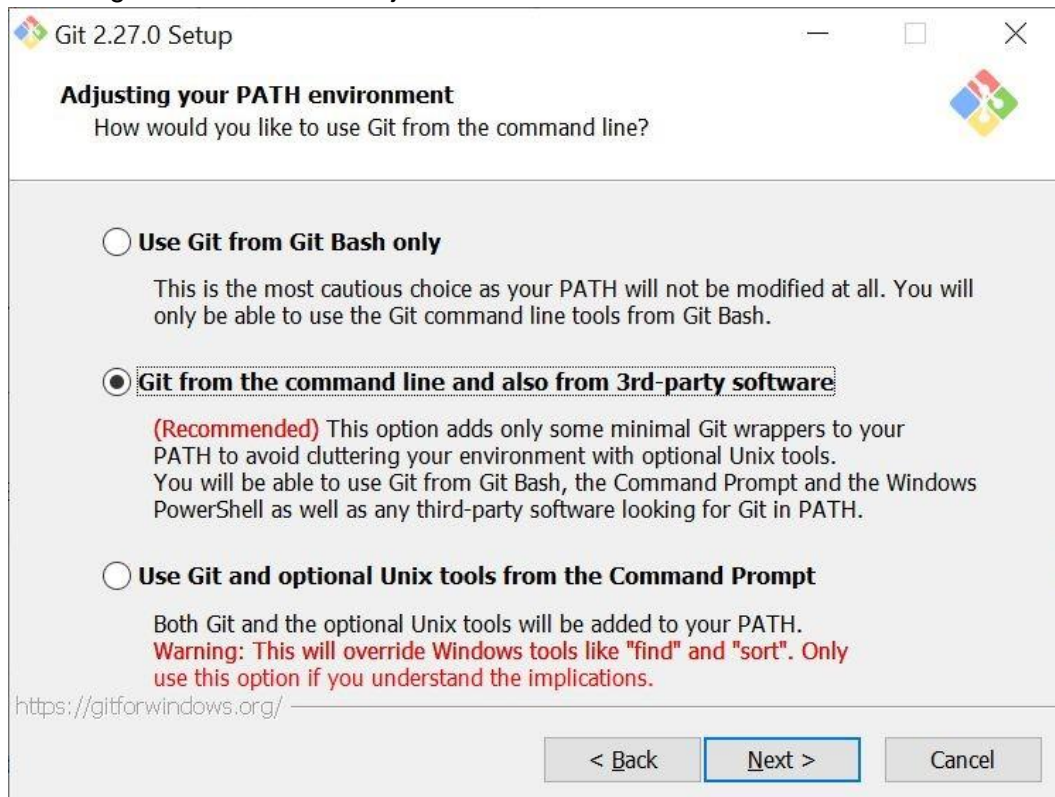


5. Seleccionar el menú de inicio (No hacer nada).

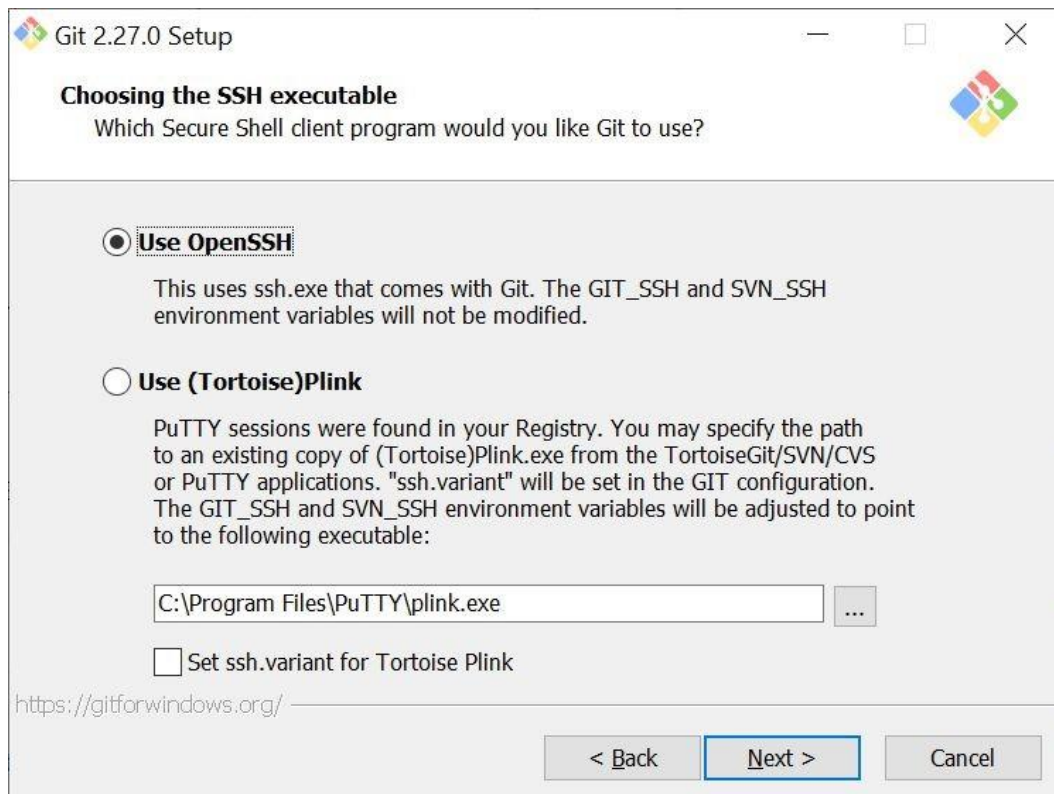
6. Seleccionar el editor de texto por defecto. Se recomienda Notepad++ o cualquier otro editor que esté instalado.



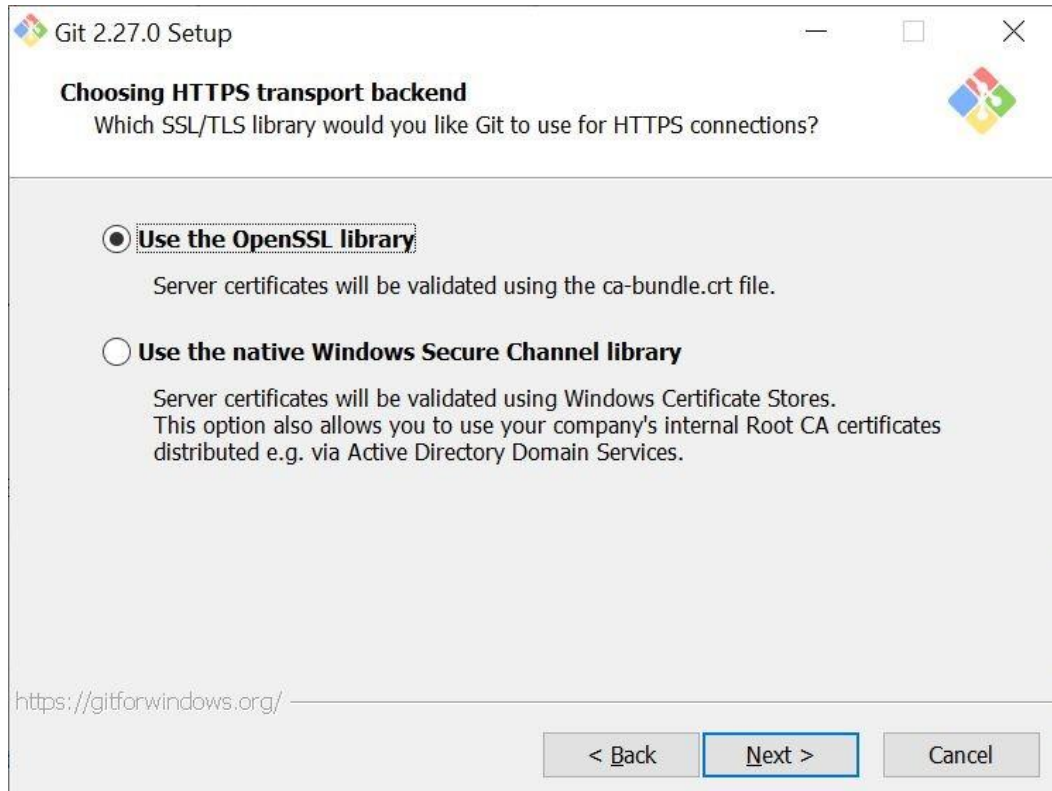
7. Configuración del Path. Dejar como lo recomienda.



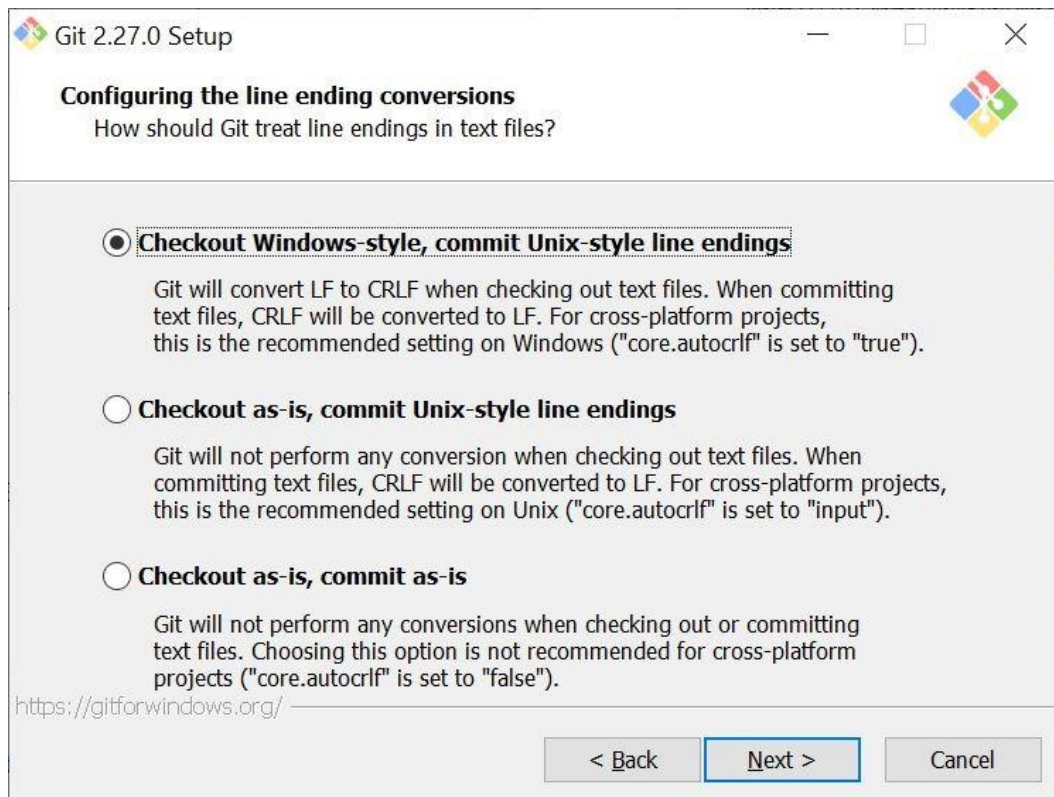
8. Configuración del SSH. Dejar la opción por defecto.



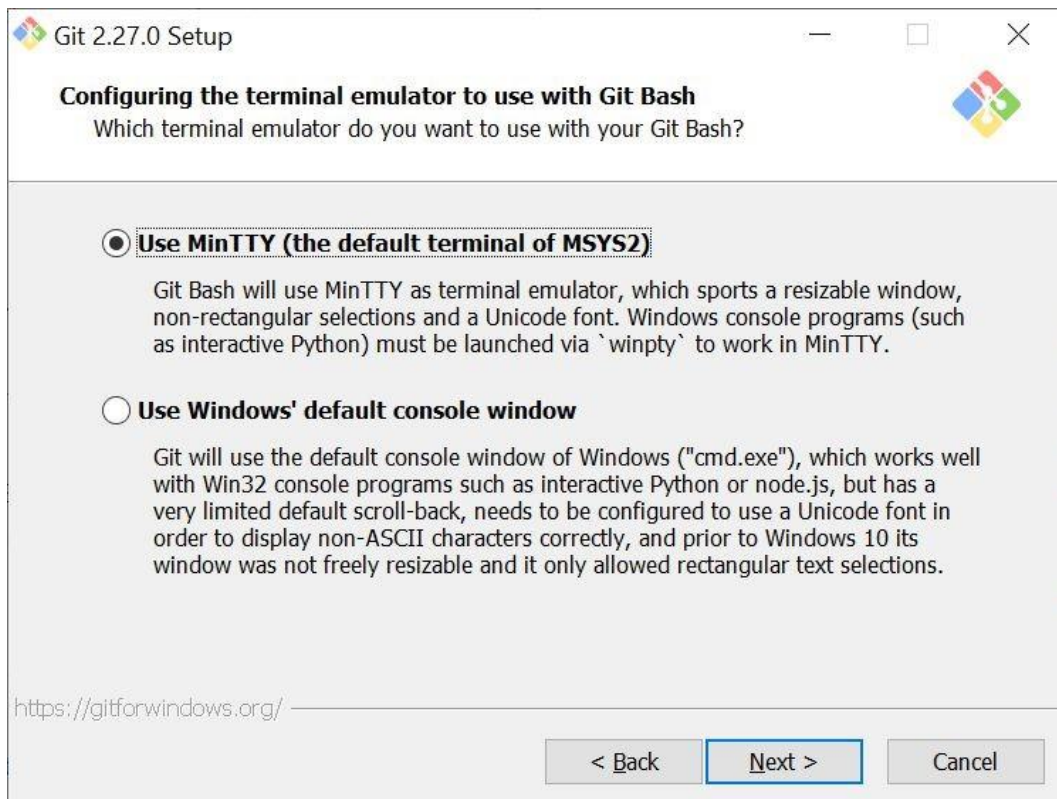
9. Utilizar SSL. Dejar por defecto.



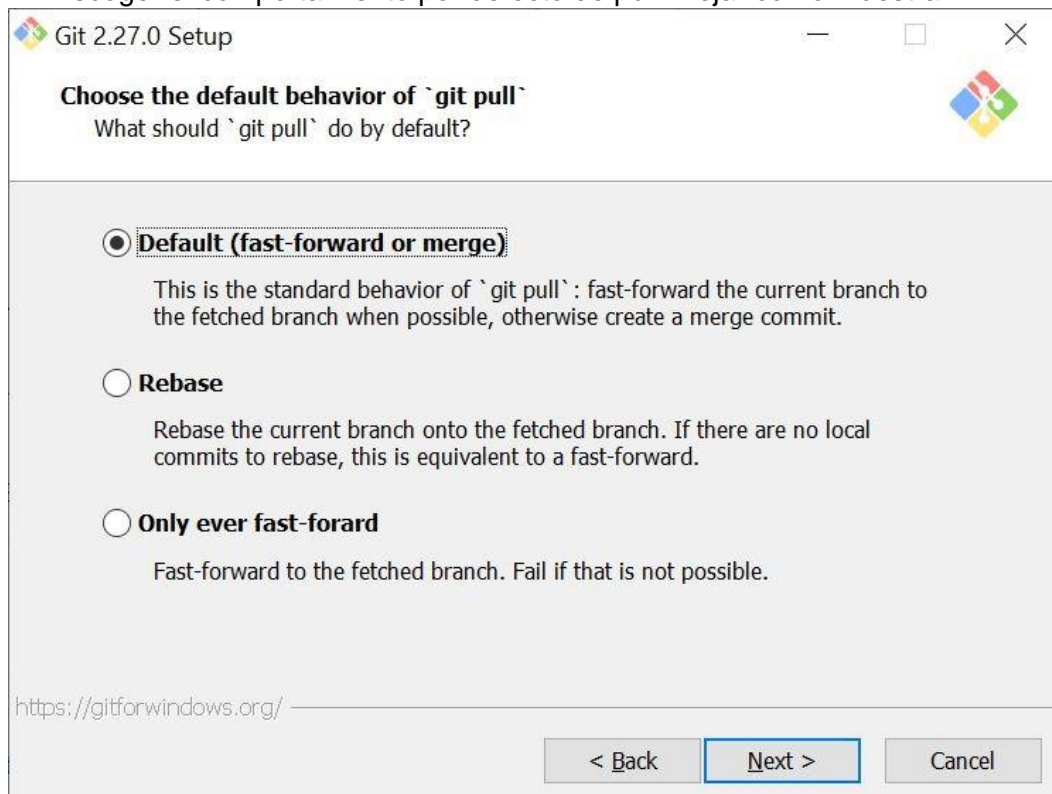
10. Configurar el salto de línea. Dejar la opción recomendada.



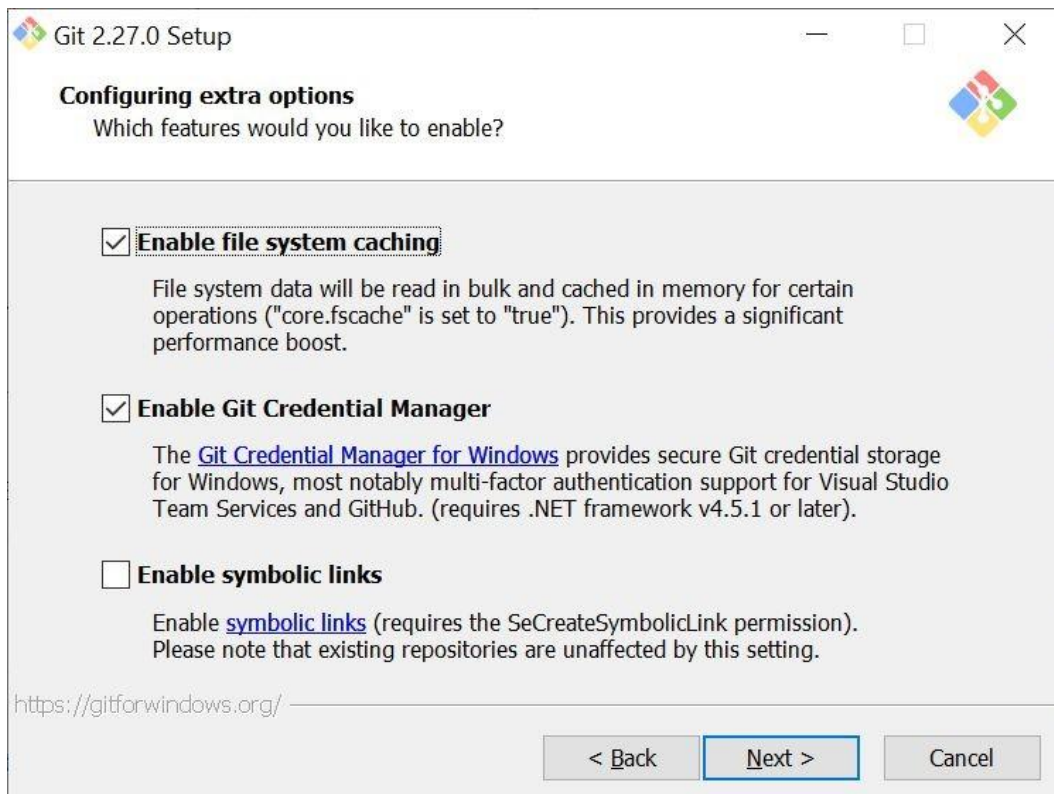
11. Configurar el Terminal. Dejar por defecto



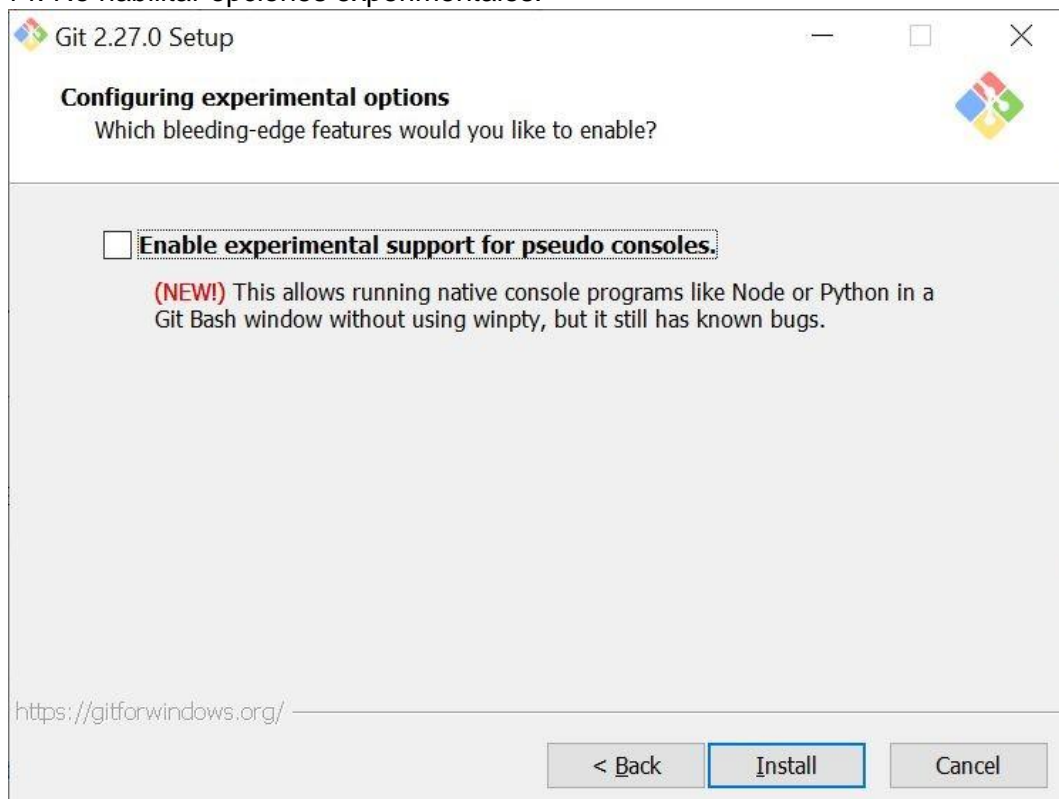
12. Escoger el comportamiento por defecto de pull. Dejar como muestra.

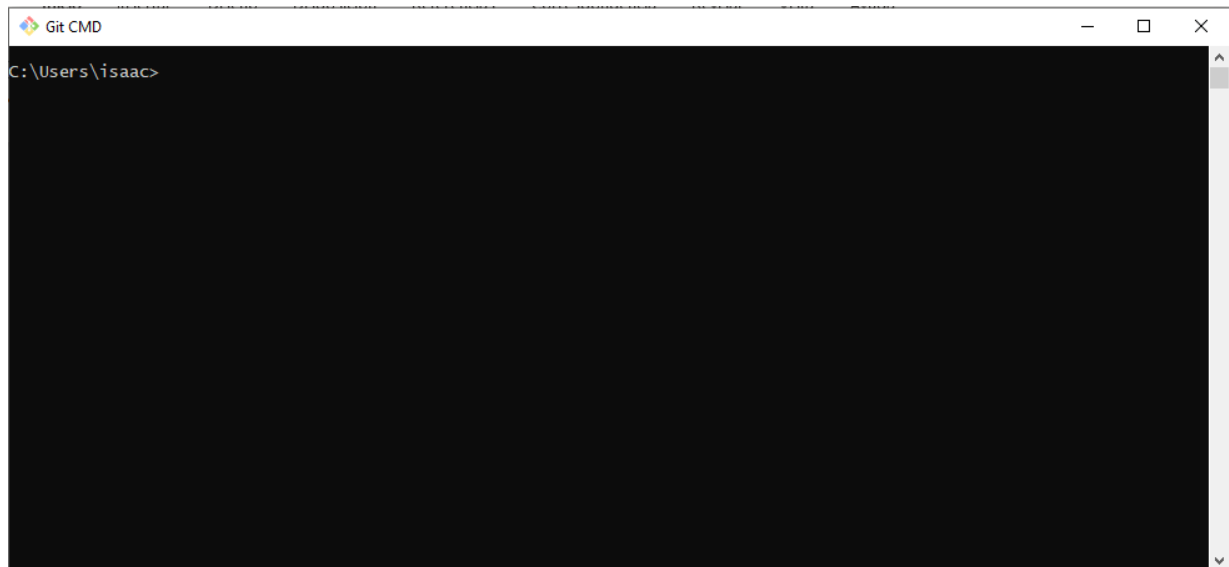


13. Configuraciones adicionales.



14. No habilitar opciones experimentales.





15. Presionar instalar.

Creando un proyecto de software

El siguiente procedimiento se podría crear utilizando las herramientas habituales de Windows o líneas de comando.

1. Abrir el **PowerShell**.

2. Crear un directorio denominado "mi-proyecto"

```
> mkdir mi-proyecto
```

3. Cambiarse a "mi-proyecto"

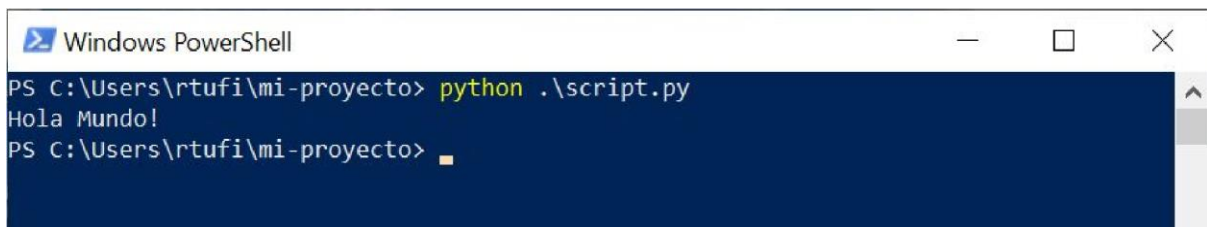
```
> cd mi-proyecto
```

4. Crear un archivo de código fuente. Para la prueba crearemos un archivo de Python denominado "script.py". Se podría utilizar cualquier editor de texto y almacenarlo dentro de la carpeta que se ha creado.

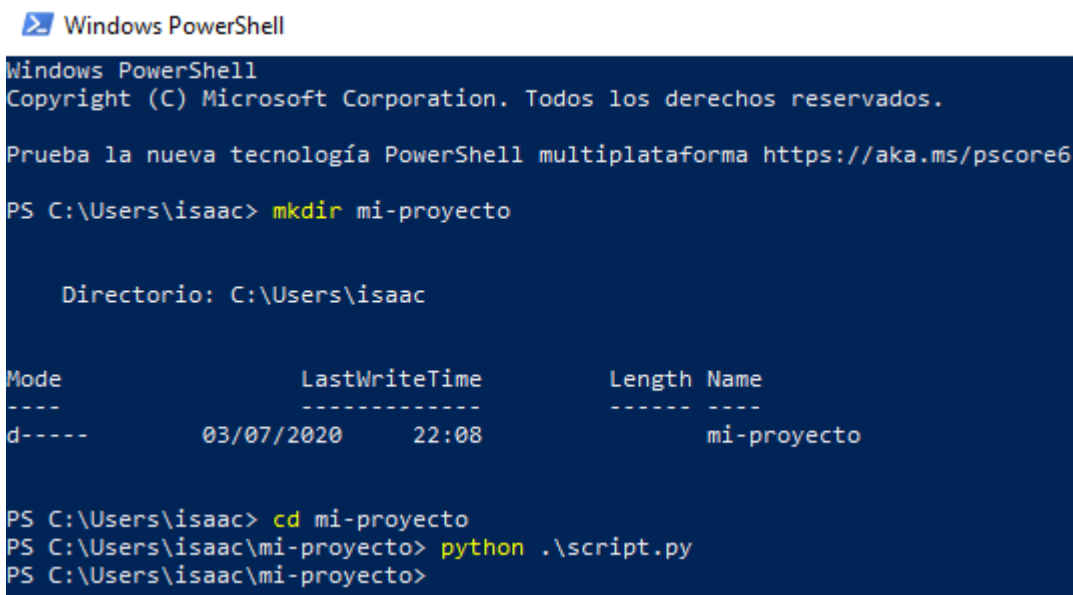

```
# Script de prueba para Git # Autor: Rodrigo
Tufiño <rtufino@ups.edu.ec>
# Fecha: 23-06-2020
# Versión: 1.0.0

def main():
    print("Hola Mundo!")
    if __name__ ==
"__main__":
        main()
```

Nota: Si se desea probar el código, primero instale Python desde la tienda **Microsoft Store**. Luego se puede realizar desde la línea de comando o el PowerShell.



```
Windows PowerShell
PS C:\Users\rtufi\mi-proyecto> python .\script.py
Hola Mundo!
PS C:\Users\rtufi\mi-proyecto> █
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\isaac> mkdir mi-proyecto

Directorio: C:\Users\isaac

Mode                LastWriteTime         Length Name
----                -
d-----          03/07/2020   22:08             mi-proyecto

PS C:\Users\isaac> cd mi-proyecto
PS C:\Users\isaac\mi-proyecto> python .\script.py
PS C:\Users\isaac\mi-proyecto>
```

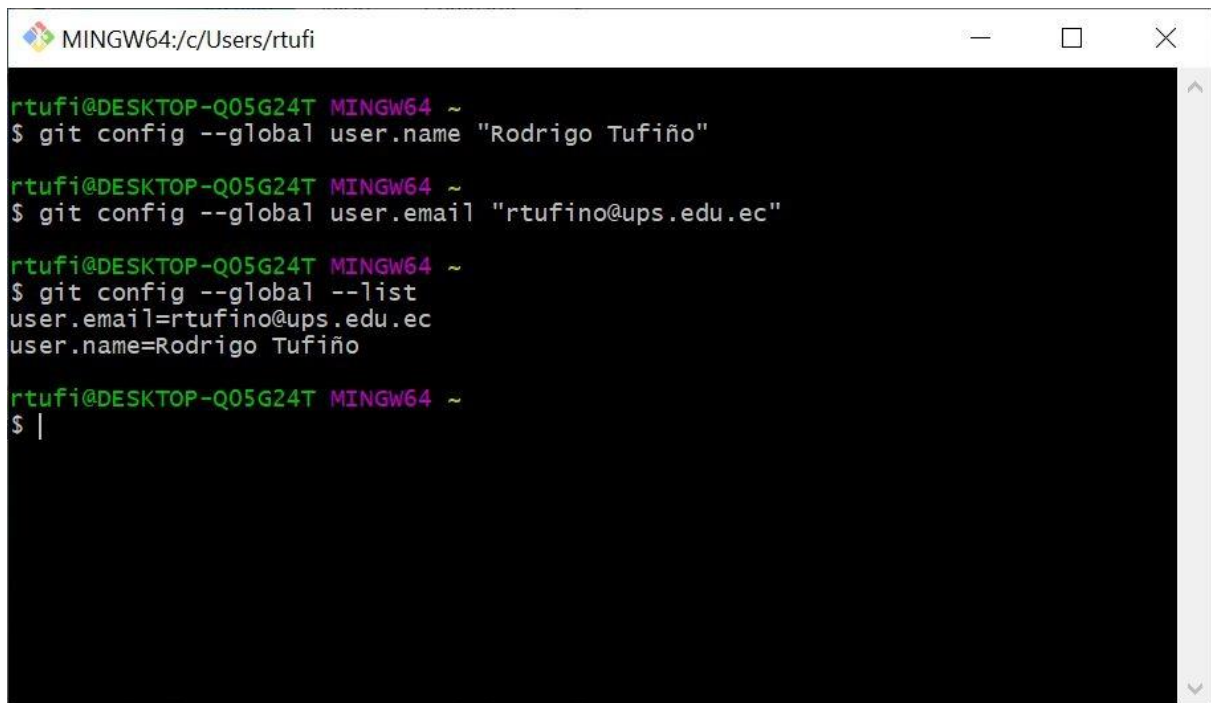
Configuración del repositorio

1. Abrir el programa **Git Bash** .

2. Cambiar de directorio a "mi-proyecto"

```
$ cd mi-proyecto/
```

3. Configurar de forma global el usuario y el correo. Verificar.

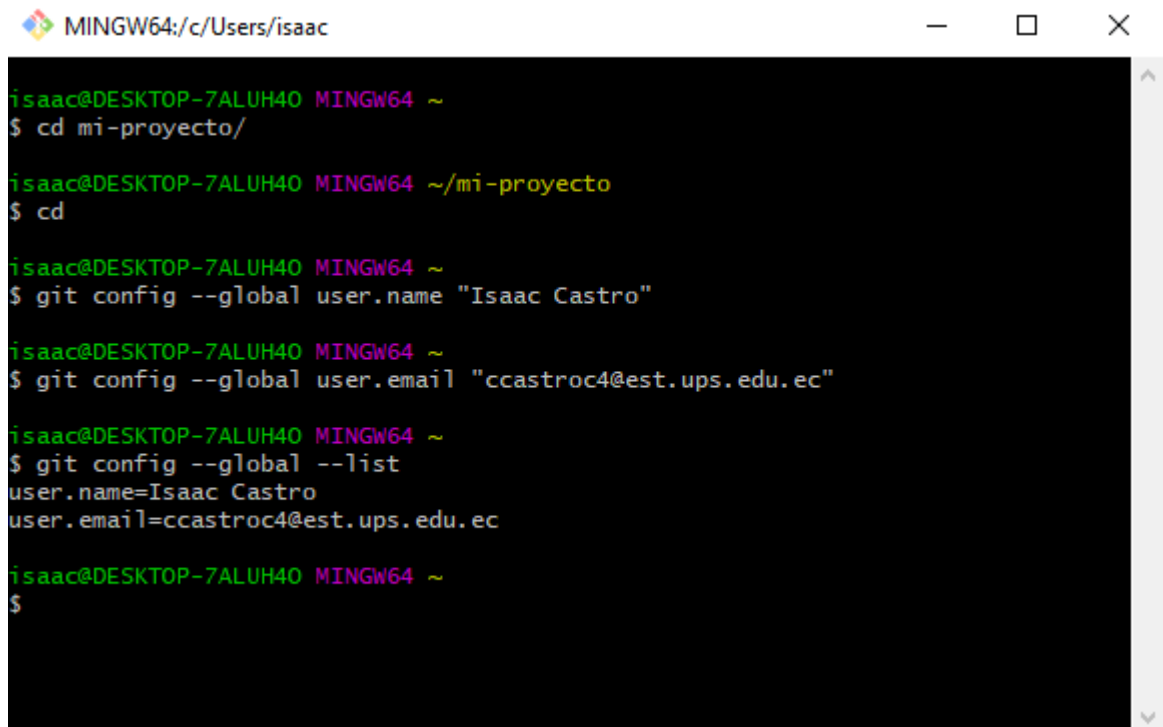


```
MINGW64:/c/Users/rtufi
rtufi@DESKTOP-Q05G24T MINGW64 ~
$ git config --global user.name "Rodrigo Tufiño"

rtufi@DESKTOP-Q05G24T MINGW64 ~
$ git config --global user.email "rtufino@ups.edu.ec"

rtufi@DESKTOP-Q05G24T MINGW64 ~
$ git config --global --list
user.email=rtufino@ups.edu.ec
user.name=Rodrigo Tufiño

rtufi@DESKTOP-Q05G24T MINGW64 ~
$ |
```



```
MINGW64:/c/Users/isaac
isaac@DESKTOP-7ALUH40 MINGW64 ~
$ cd mi-proyecto/

isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto
$ cd

isaac@DESKTOP-7ALUH40 MINGW64 ~
$ git config --global user.name "Isaac Castro"

isaac@DESKTOP-7ALUH40 MINGW64 ~
$ git config --global user.email "ccastroc4@est.ups.edu.ec"

isaac@DESKTOP-7ALUH40 MINGW64 ~
$ git config --global --list
user.name=Isaac Castro
user.email=ccastroc4@est.ups.edu.ec

isaac@DESKTOP-7ALUH40 MINGW64 ~
$
```

Trabajando con el repositorio local

1. Con el **Git Bash** , dirigirse al directorio del proyecto

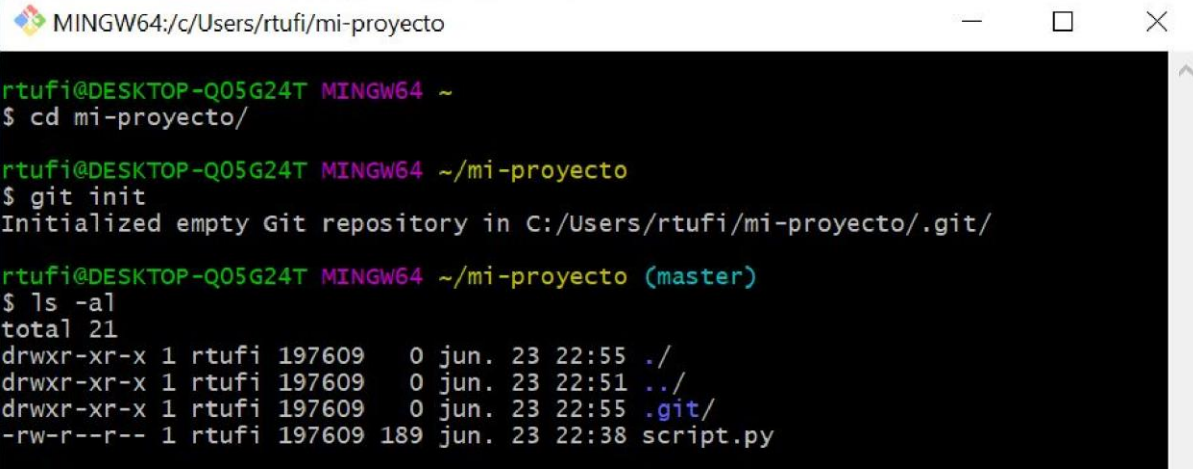
```
$ cd mi-proyecto
```

2. Crear el repositorio

```
$ git init
```

3. Verificar que el directorio oculto se ha creado

```
$ ls -al
```



The screenshot shows a Windows terminal window titled 'MINGW64:/c/Users/rtufi/mi-proyecto'. The user 'rtufi' runs the following commands: `cd mi-proyecto/`, `git init`, and `ls -al`. The output shows the creation of a '.git/' directory and a 'script.py' file. Below this, another terminal window for user 'isaac' is shown, where the same commands are executed, resulting in a similar directory structure but with a different file size for 'script.py' (206 bytes).

```
rtufi@DESKTOP-Q05G24T MINGW64 ~  
$ cd mi-proyecto/  
  
rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto  
$ git init  
Initialized empty Git repository in C:/Users/rtufi/mi-proyecto/.git/  
  
rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)  
$ ls -al  
total 21  
drwxr-xr-x 1 rtufi 197609  0 jun. 23 22:55 ./  
drwxr-xr-x 1 rtufi 197609  0 jun. 23 22:51 ../  
drwxr-xr-x 1 rtufi 197609  0 jun. 23 22:55 .git/  
-rw-r--r-- 1 rtufi 197609 189 jun. 23 22:38 script.py  
  
isaac@DESKTOP-7ALUH40 MINGW64 ~  
$ cd mi-proyecto  
  
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto  
$ git init  
Initialized empty Git repository in C:/Users/isaac/mi-proyecto/.git/  
  
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)  
$ ls -al  
total 21  
drwxr-xr-x 1 isaac 197609  0 jul.  3 22:12 ./  
drwxr-xr-x 1 isaac 197609  0 jul.  3 22:12 ../  
drwxr-xr-x 1 isaac 197609  0 jul.  3 22:12 .git/  
-rw-r--r-- 1 isaac 197609 206 jul.  3 22:09 script.py  
  
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
```

4. Verificar el estado del repositorio

```
$ git status
```

```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        script.py

nothing added to commit but untracked files present (use "git add" to track)
```

5. Agregar el script.py a la zona de stage

```
$ git add script.py
```

```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git add script.py
warning: LF will be replaced by CRLF in script.py.
The file will have its original line endings in your working directory
```

6. Volver a verificar el estado del repositorio

```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   script.py
```

```
MINGW64:/c/Users/isaac/mi-proyecto
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        script.py

nothing added to commit but untracked files present (use "git add" to track)

isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git add script.py

isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   script.py

isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
```

7. Comprometer el proyecto

```
$ git commit -m "Commit inicial"
```

```
MINGW64:/c/Users/rtufi/mi-proyecto
rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git commit -m "Commit inicial"
[master (root-commit) 550d6df] Commit inicial
1 file changed, 11 insertions(+)
create mode 100644 script.py
```

8. Volver a verificar el estado del repositorio

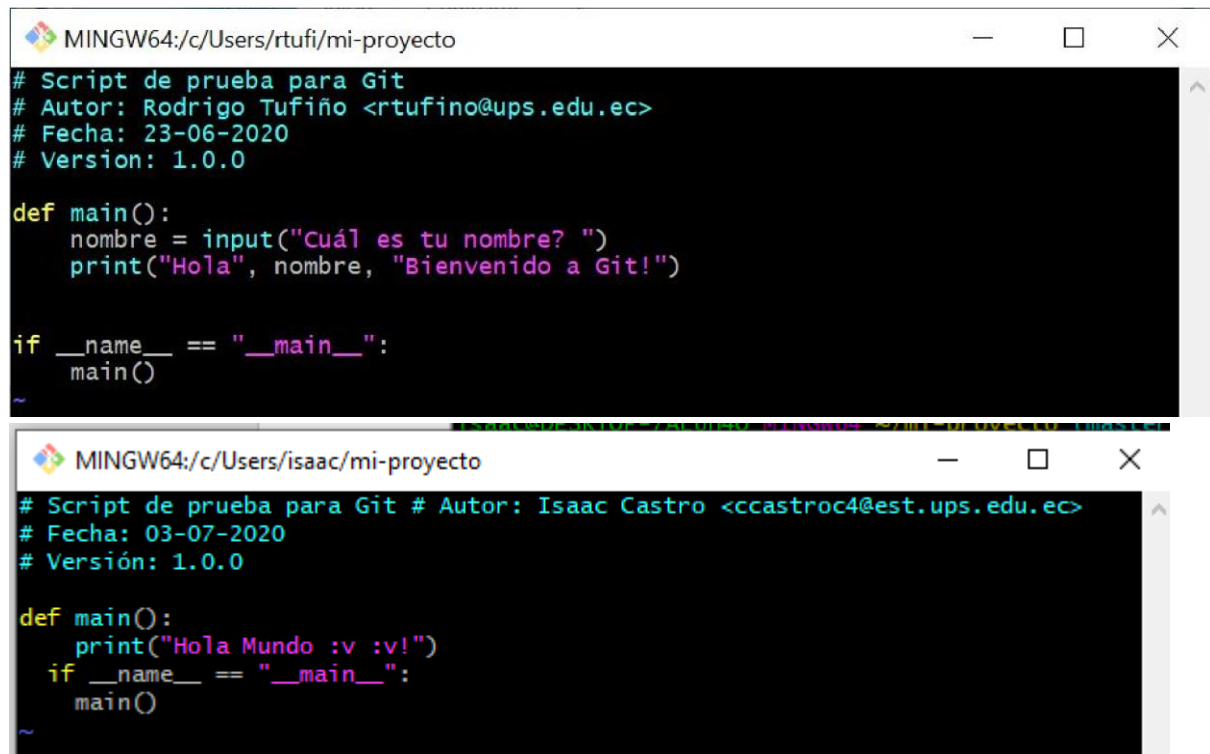
```
MINGW64:/c/Users/rtufi/mi-proyecto
rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git commit -m "Commit inicial"
[master (root-commit) e5e925d] Commit inicial
1 file changed, 8 insertions(+)
create mode 100644 script.py

isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git status
On branch master
nothing to commit, working tree clean

isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$
```

9. Modificar el script. Se lo puede hacer con cualquier editor de texto. Para este caso se ha utilizado **vi** . Guardar los cambios realizados.



```
MINGW64:/c/Users/rtufi/mi-proyecto
# Script de prueba para Git
# Autor: Rodrigo Tufiño <rtufino@ups.edu.ec>
# Fecha: 23-06-2020
# Version: 1.0.0

def main():
    nombre = input("Cuál es tu nombre? ")
    print("Hola", nombre, "Bienvenido a Git!")

if __name__ == "__main__":
    main()

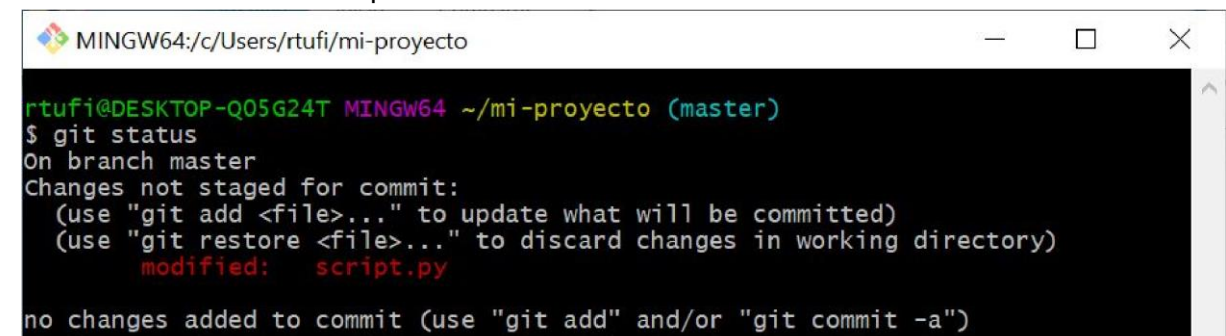
~
```

```
MINGW64:/c/Users/isaac/mi-proyecto
# Script de prueba para Git # Autor: Isaac Castro <ccastroc4@est.ups.edu.ec>
# Fecha: 03-07-2020
# Versión: 1.0.0

def main():
    print("Hola Mundo :v :v!")
    if __name__ == "__main__":
        main()

~
```

10. Verificar el estado del repositorio



```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   script.py

no changes added to commit (use "git add" and/or "git commit -a")
```



```
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   script.py

no changes added to commit (use "git add" and/or "git commit -a")
```

11. Crear un nuevo archivo denominado "datos.csv" con un par de líneas de datos.

```
MINGW64:/c/Users/rtufi/mi-proyecto
1;"Pepito Perez";78
2;"Juan Obando";86
3;"Camila Cedeño";94
```

```
MINGW64:/c/Users/isaac/mi-proyecto
1;"Pepito Perez";78
2;"Juan Obando";78
3;"Camila Camila";45
```

12. Volver a verificar el estado del repositorio

```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   script.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        datos.csv

no changes added to commit (use "git add" and/or "git commit -a")
```

```
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   script.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        datos.csv

no changes added to commit (use "git add" and/or "git commit -a")
```

13. Agregar los archivos al área de stage

```
$ git add .
```

```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git add .
warning: LF will be replaced by CRLF in script.py.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in datos.csv.
The file will have its original line endings in your working directory
```

```
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git add .
warning: LF will be replaced by CRLF in datos.csv.
The file will have its original line endings in your working directory
```

14. Volver a verificar el estado del repositorio

```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   datos.csv
    modified:   script.py
```

```
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   datos.csv
    modified:   script.py
```

15. Comprometer los cambios realizados

```
$ git commit -m "Agregado datos del proyecto"
```

```
MINGW64:/c/Users/rtufi/mi-proyecto

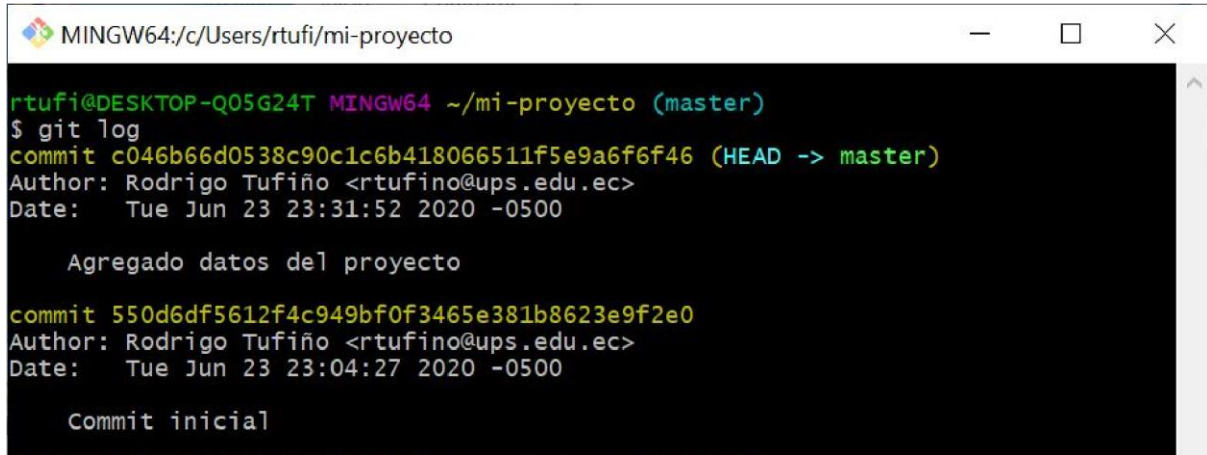
rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git commit -m "Agregado datos del proyecto"
[master c046b66] Agregado datos del proyecto
 2 files changed, 5 insertions(+), 1 deletion(-)
 create mode 100644 datos.csv
```

```
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git commit -m "Agregado datos del proyecto"
[master e2709a7] Agregado datos del proyecto
 2 files changed, 6 insertions(+), 3 deletions(-)
 create mode 100644 datos.csv
```

Revisando commits realizados

1. Revisar la lista de commits en la rama principal

\$ git log



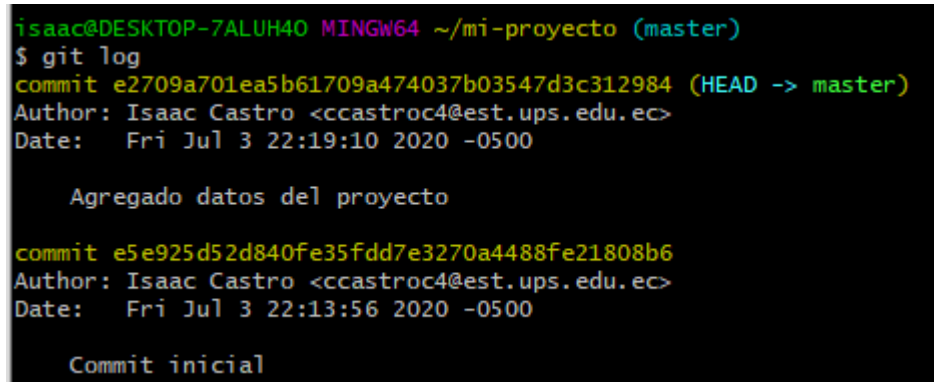
```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git log
commit c046b66d0538c90c1c6b418066511f5e9a6f6f46 (HEAD -> master)
Author: Rodrigo Tufiño <rtufino@ups.edu.ec>
Date: Tue Jun 23 23:31:52 2020 -0500

    Agregado datos del proyecto

commit 550d6df5612f4c949bf0f3465e381b8623e9f2e0
Author: Rodrigo Tufiño <rtufino@ups.edu.ec>
Date: Tue Jun 23 23:04:27 2020 -0500

    Commit inicial
```



```
isaac@DESKTOP-7ALUH40 MINGW64 ~/mi-proyecto (master)
$ git log
commit e2709a701ea5b61709a474037b03547d3c312984 (HEAD -> master)
Author: Isaac Castro <ccastroc4@est.ups.edu.ec>
Date: Fri Jul 3 22:19:10 2020 -0500

    Agregado datos del proyecto

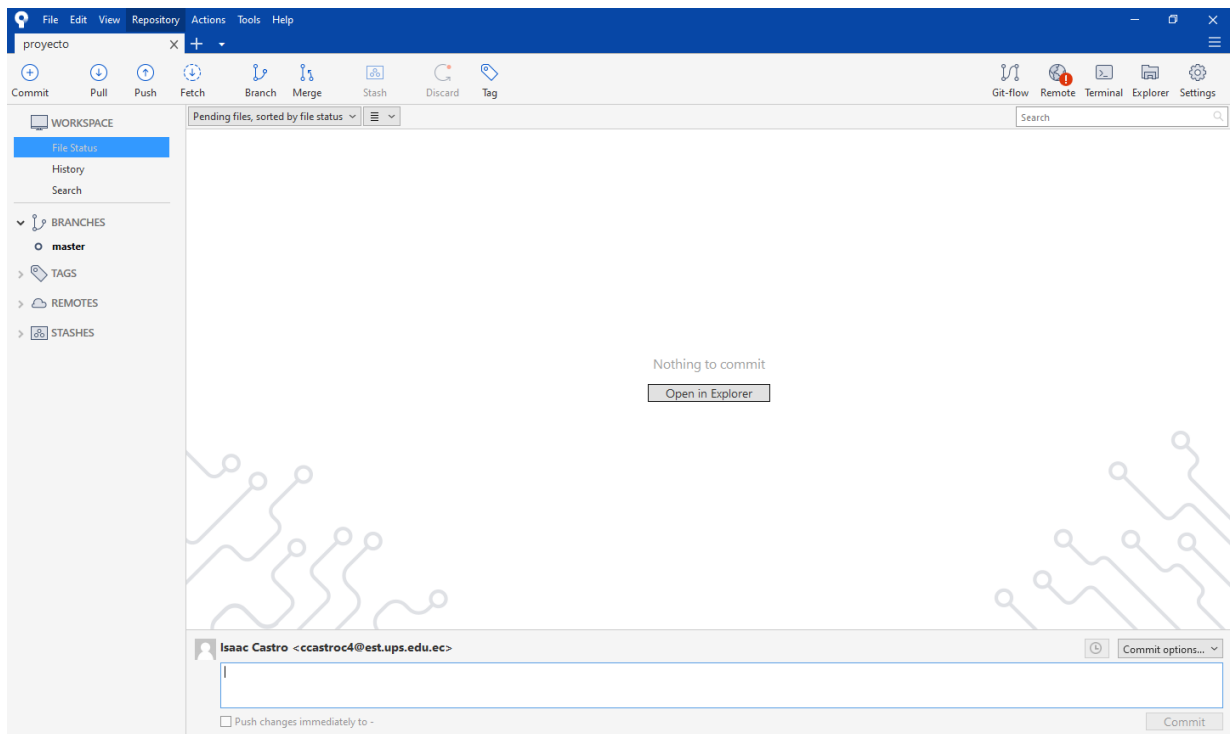
commit e5e925d52d840fe35fdd7e3270a4488fe21808b6
Author: Isaac Castro <ccastroc4@est.ups.edu.ec>
Date: Fri Jul 3 22:13:56 2020 -0500

    Commit inicial
```

2. Instalar el programa **SourceTree** .

- Ignorar el registro en Atlassian
- Seleccionar Mercurial
- Configurar las preferencias con las mismas credenciales que en Git.
- Sobre la clave SSH, presionar NO

3. Abrir el programa **Sourcetree** .



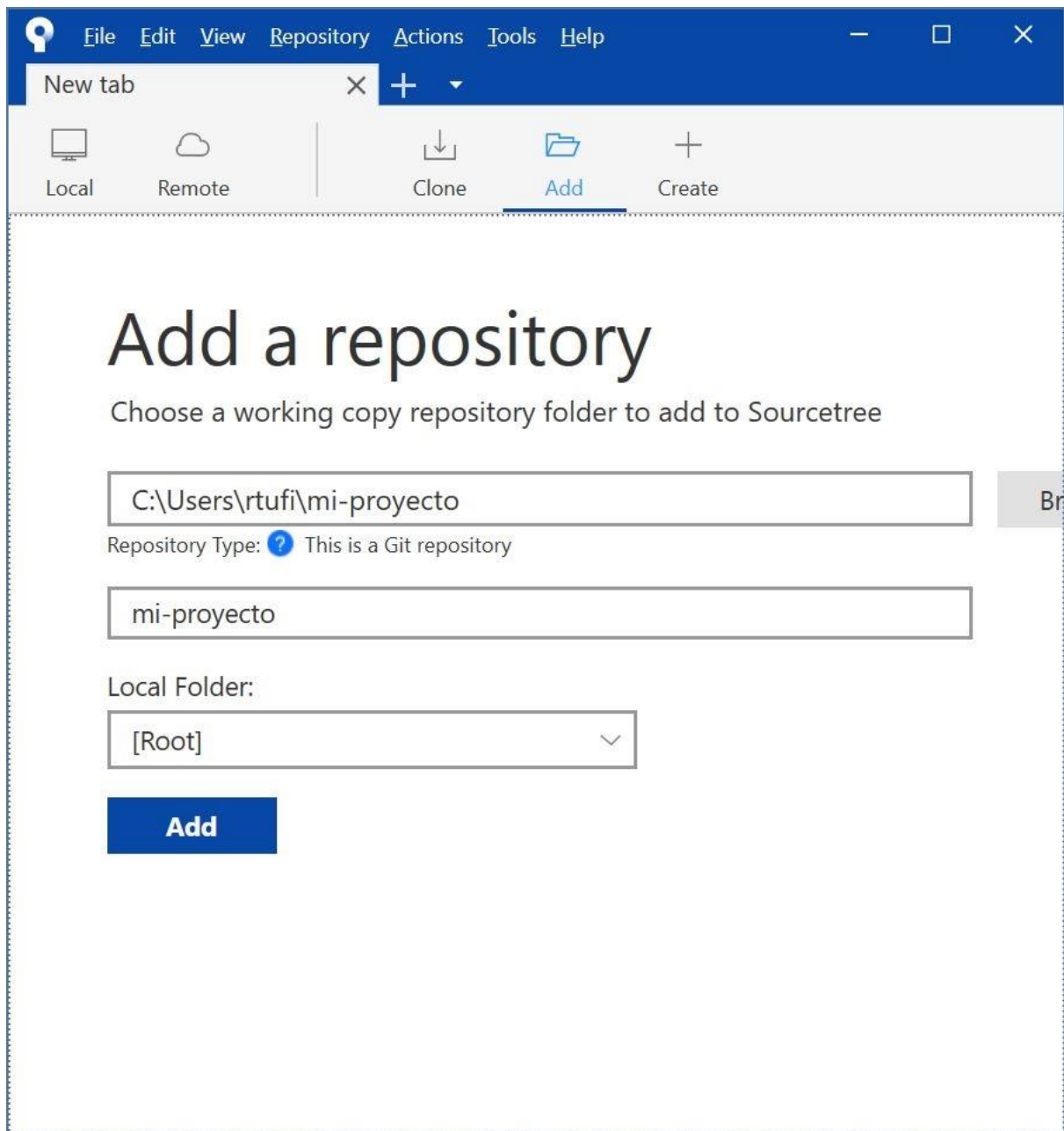
4. Presionar en el ícono “Add”. Explorar hasta la carpeta del proyecto y presionar el botón “Add”.

Add a repository

Choose a working copy repository folder to add to Sourcetree

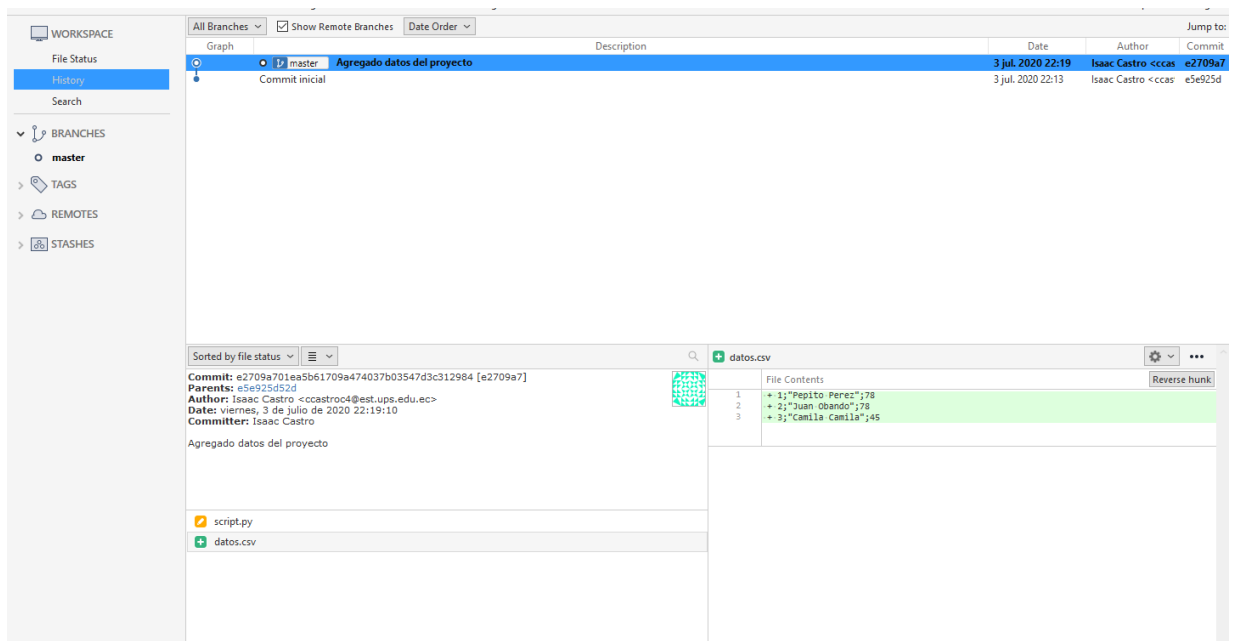
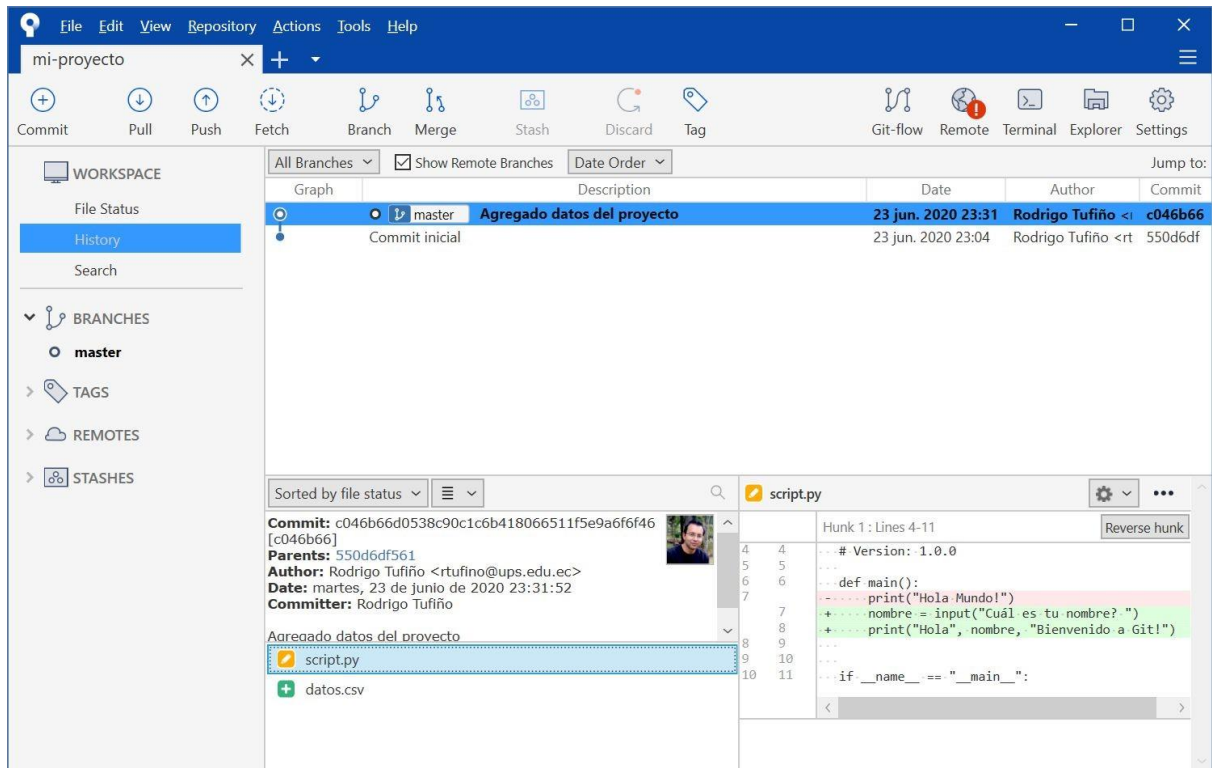
Repository Type: This is a Git repository

Local Folder:



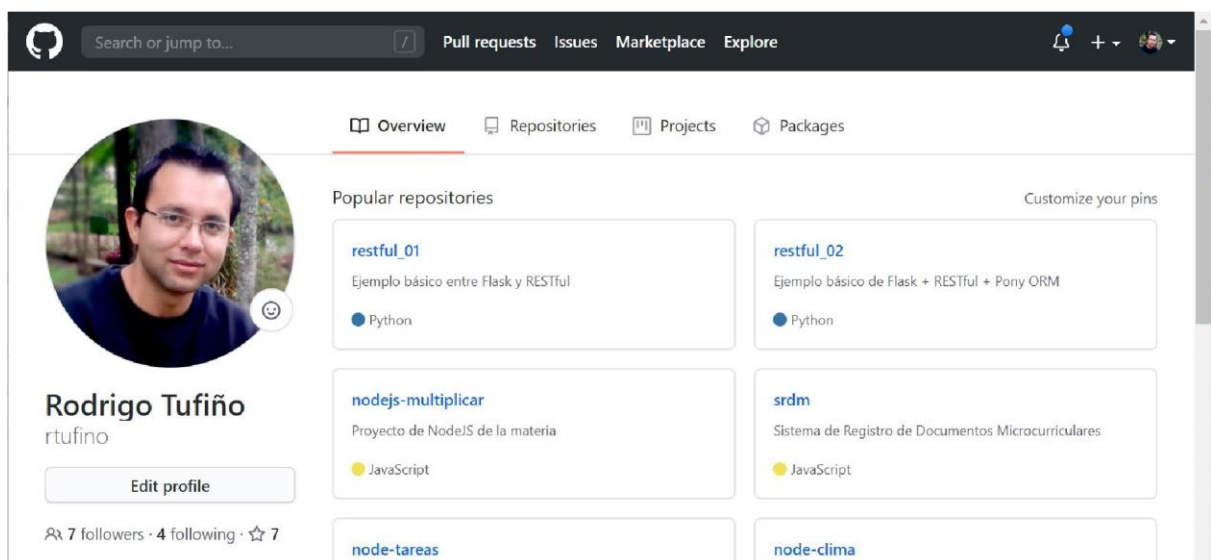
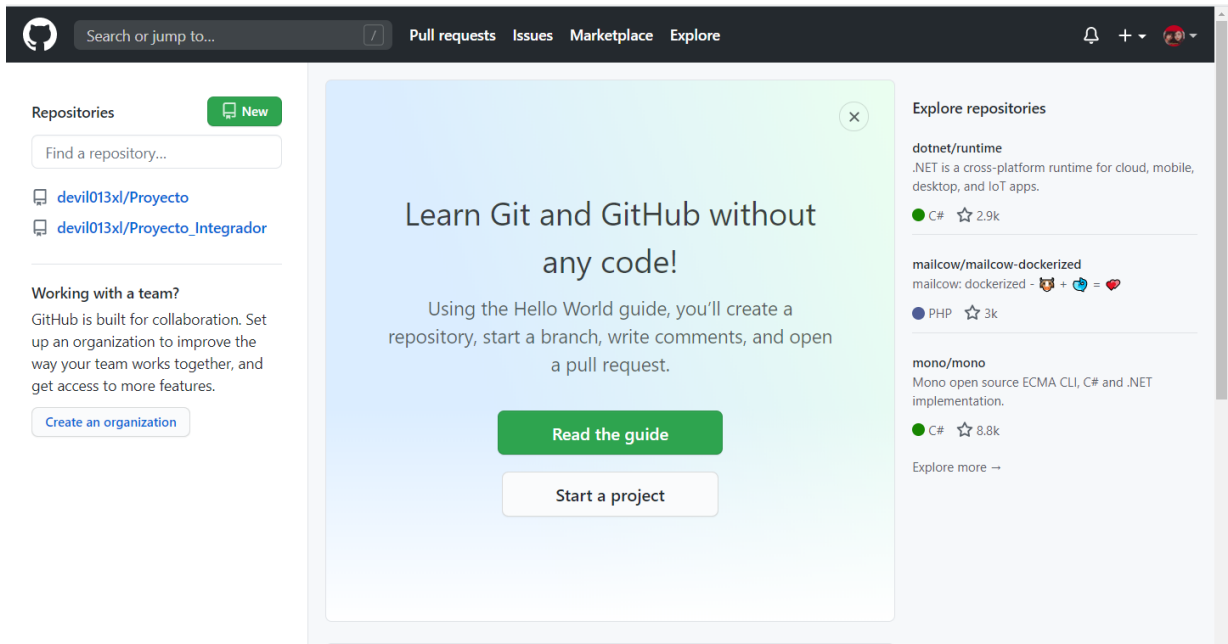
5. El programa mostrará información del repositorio de una manera más intuitiva y fácil de entender. Se puede apreciar los commits realizados y los detalles de cada uno de ellos. El autor y los cambios que han sido realizados sobre los archivos.

Nota: Esta herramienta no solo permite visualizar el estado del repositorio, sino que también permite trabajar sobre el repositorio local y remoto

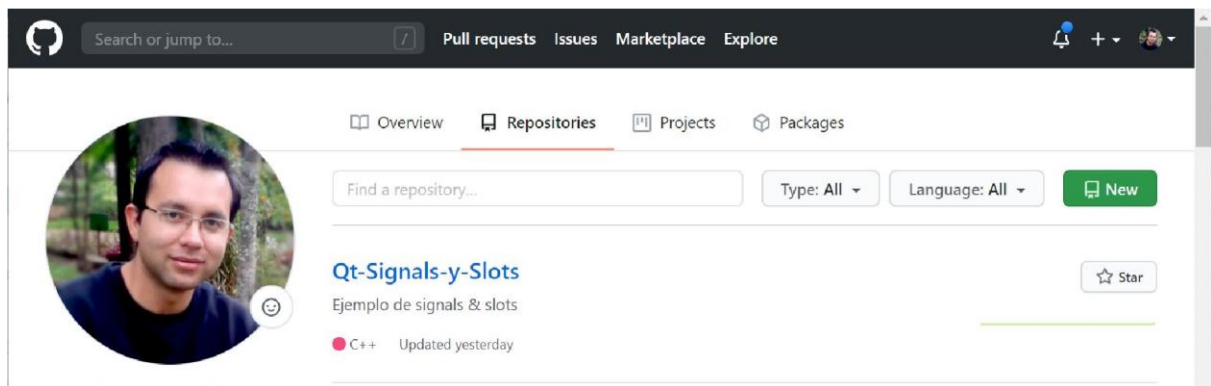


Trabajando con repositorios remotos

1. Crear y configurar una cuenta en [GitHub](#). Se recomienda colocar una foto de perfil y completar el perfil



2. Crear un repositorio: **Repositories > New**

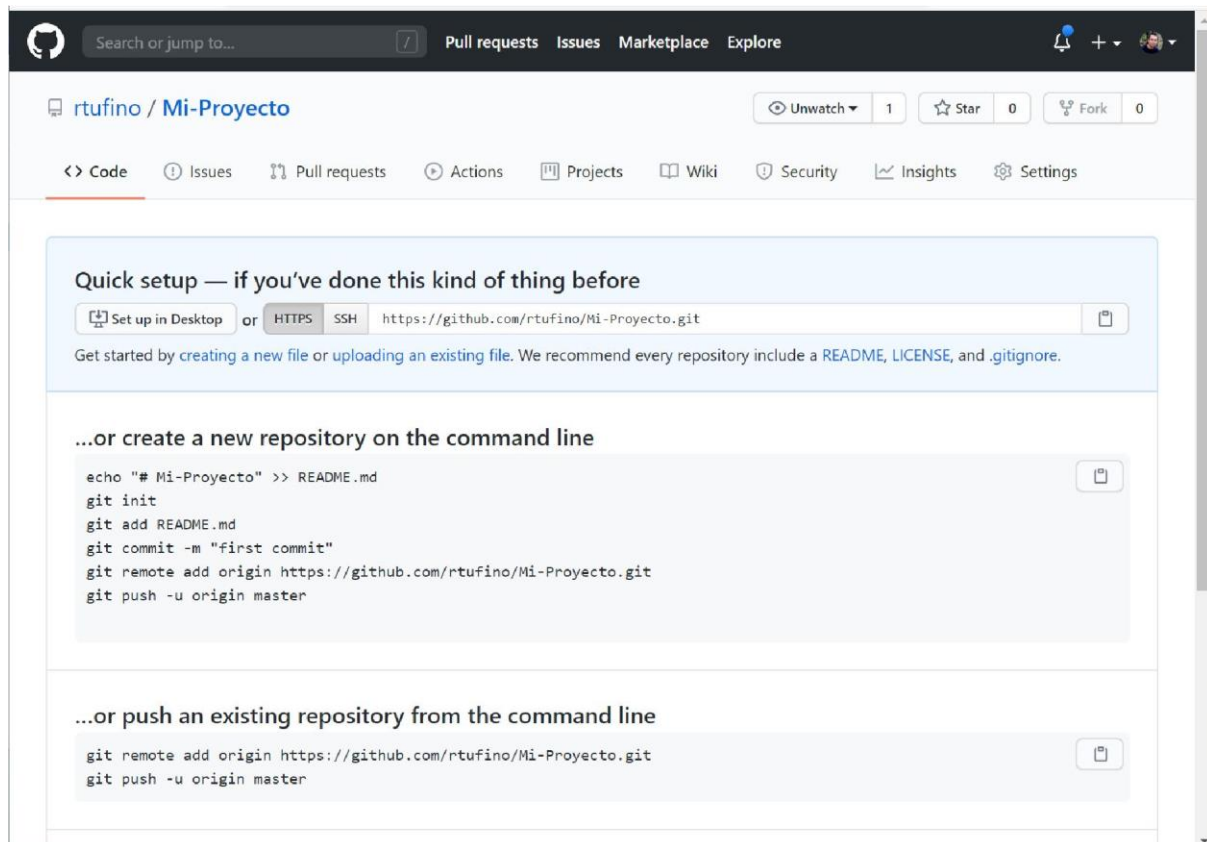


3. Colocar los datos para el repositorio. Por ejemplo:

- **Nombre:** Mi-Proyecto (No es obligatorio que se llame como la carpeta del proyecto)
- **Descripción:** Un proyecto de ejemplo para GitHub
- **Público**

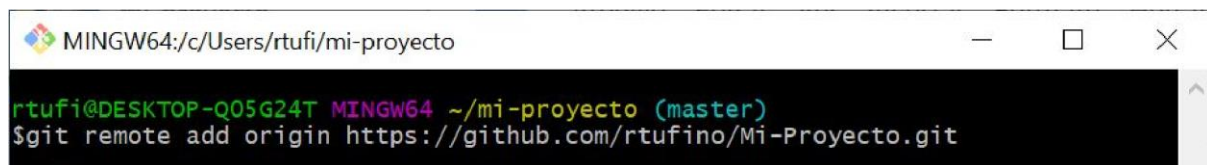
 A screenshot of the 'Create a new repository' form on GitHub. The title 'Create a new repository' is at the top, followed by a subtitle explaining that a repository contains all project files and revision history. Below this, there's a section for 'Owner' and 'Repository name'. The owner is 'rtufino' and the repository name is 'Mi-Proyecto', which has a green checkmark next to it. A note suggests that great repository names are short and memorable. The 'Description (optional)' field contains the text 'Un proyecto de ejemplo para GitHub'. There are two radio buttons for visibility: 'Public' (selected) and 'Private'. The 'Public' option is described as allowing anyone on the internet to see the repository. Below this, there's a checkbox for 'Initialize this repository with a README', which is currently unchecked. A note explains that this will let you immediately clone the repository to your computer. At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by an information icon. A large green 'Create repository' button is at the very bottom.

4. Al momento de crear el repositorio (Create repository) aparecerá la siguiente información:



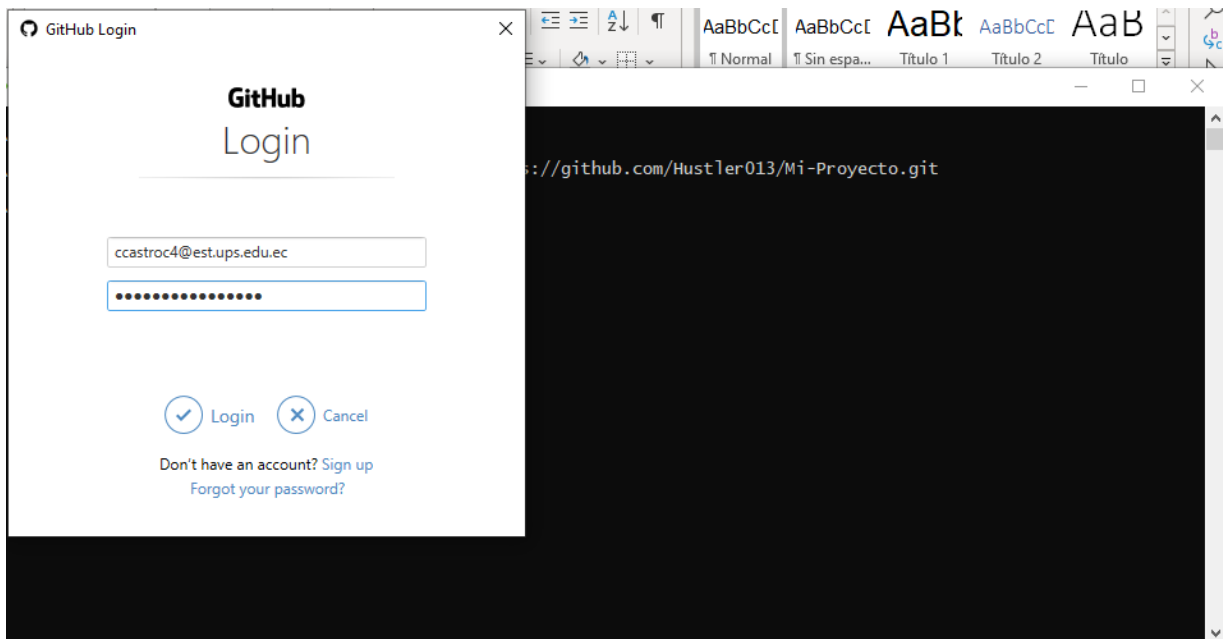
5. Conectar el repositorio remoto con el repositorio local. Dentro del Git Bash copiar la sentencia que proporciona GitHub

```
$ git remote add origin https://github.com/rtufino/Mi-Proyecto.git
```



6. Actualizar las referencias remotas con la información local. Dentro del Git Bash copiar la sentencia que proporciona GitHub

```
$ git push -u origin master
```



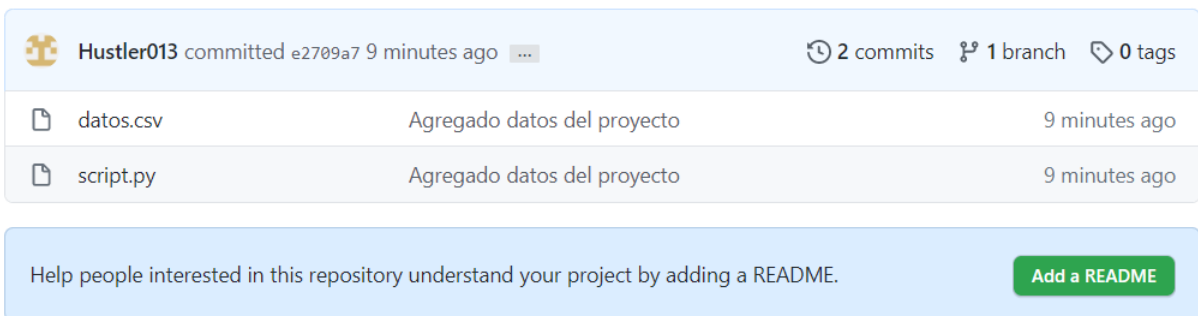
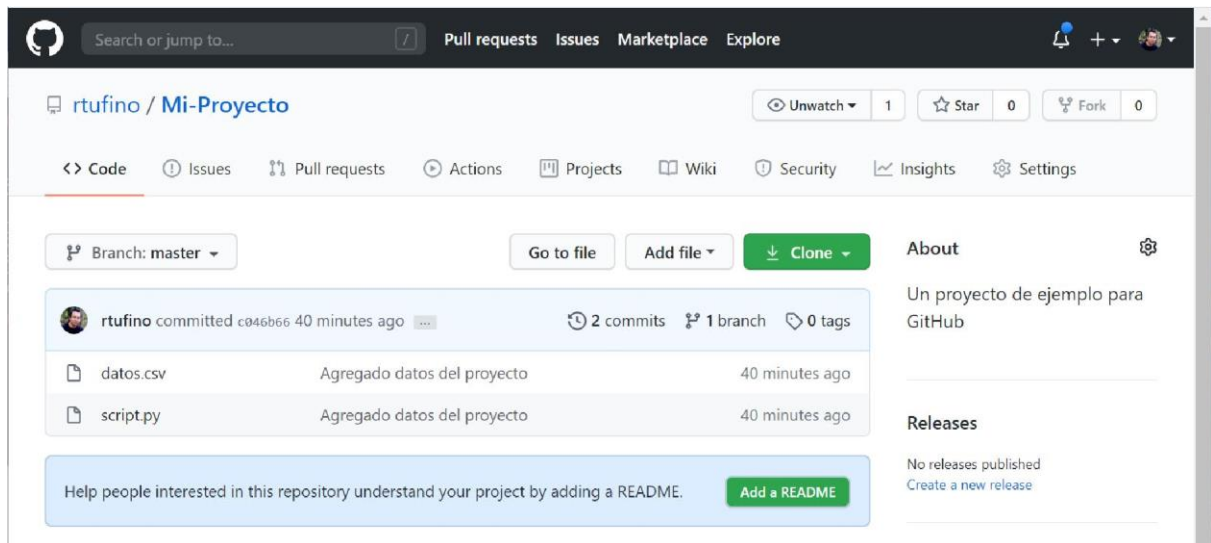
Nota: Solo la primera vez se debe colocar la opción -u origin master. Para las siguientes actualizaciones basta con ejecutar git push.

```
MINGW64:/c/Users/rtufi/mi-proyecto

rtufi@DESKTOP-Q05G24T MINGW64 ~/mi-proyecto (master)
$ git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 787 bytes | 196.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/rtufino/Mi-Proyecto.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 754 bytes | 377.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Hustler013/Mi-Proyecto.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

7. Actualizar la página de GitHub.

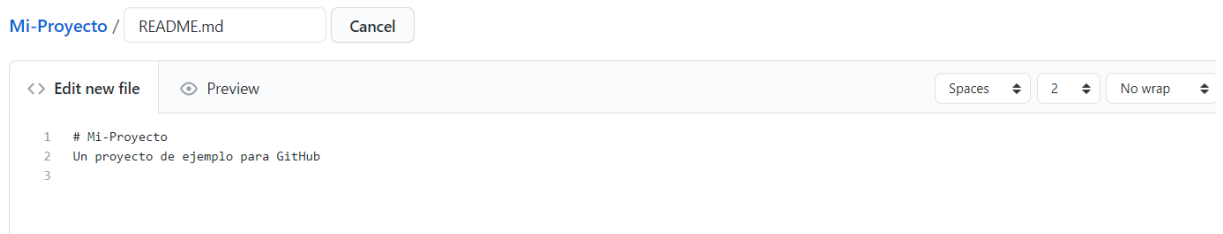


Notar que aparecen los dos commits realizados y que se puede navegar por cada uno de ellos.

Trabajo autónomo:

Investigar y probar los siguientes conceptos y funciones de Git y GitHub:

1. Crear una página de README dentro del repositorio.



2. Investigar sobre el archivo ".gitignore" y cuál es su función.



Ignorando cosas

📄 Hoja de ruta

Lección: 5 min
Prácticas: 0 min

Preguntas

- ¿Cómo puedo indicarle a Git que ignore los archivos que no quiero rastrear?

Objetivos

- Configure Git para ignorar archivos específicos.
- Explica por qué ignorar los archivos puede ser útil.

¿Qué pasa si tenemos archivos que no queremos que Git rastree, como archivos de copia de seguridad creados por nuestro editor o archivos intermedios creados durante el análisis de datos? Vamos a crear algunos archivos ficticios:

```
$ mkdir results
$ touch a.dat b.dat c.dat results/a.out results/b.out
```

Mira lo que Git dice:

```
$ git status
```

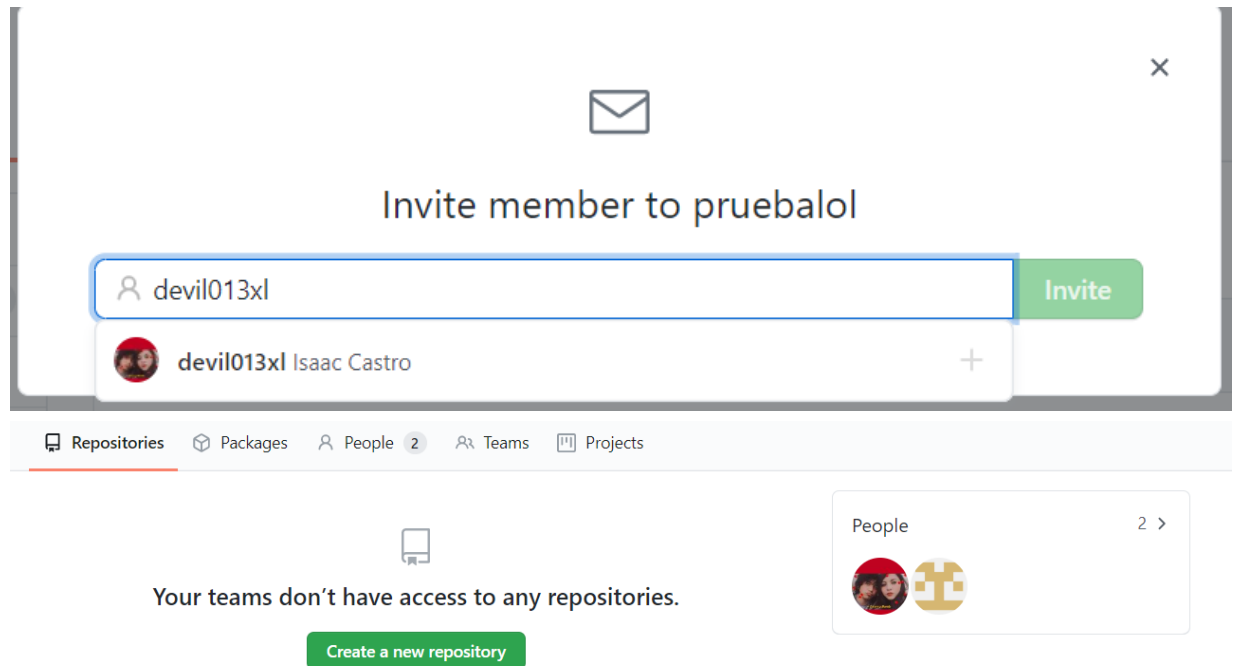
```
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    a.dat
    b.dat
    c.dat
    results/
nothing added to commit but untracked files present (use "git add" to track)
```

Colocar estos archivos bajo el control de versiones sería un desperdicio de espacio en disco. Y lo que es peor, al tenerlos todos listados, podría distraernos de los cambios que

<https://swcarpentry.github.io/git-novice-es/06-ignore/>

3. Agregar más participantes al repositorio GitHub.



Hustler013 / Mi-Proyecto Unwatch

Code Issues Pull requests Actions Projects Wiki Security Insights

Branch: isaac39@hotmail... Go to file Add file Code

This branch is even with master. Pull request Compare

Hustler013 committed 7dd12c6 1 minute ago 3 commits 2 branches 0 tags

README.md	Create README.md	1 minute ago
datos.csv	Agregado datos del proyecto	12 minutes ago
script.py	Agregado datos del proyecto	12 minutes ago

- Investigar el comando git clone.

```
C:\Users\isaac\mi-proyecto>git clone https://github.com/Hustler013/Mi-Proyecto
Cloning into 'Mi-Proyecto'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 1), reused 7 (delta 1), pack-reused 0
Unpacking objects: 100% (10/10), 1.42 KiB | 3.00 KiB/s, done.
C:\Users\isaac\mi-proyecto>
```

- Configurar el repositorio remoto de un compañero y realizar actualizaciones.

devil013xl / Proyecto

Watch0

Star0

Fork0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

You're making changes in a project you don't have write access to. We've **created a fork of this project** for you to commit your proposed changes to. Submitting a change will write it to a new branch in your fork, so you can send a pull request.

Proyecto / lol

Cancel

<> Edit new file

Preview

Spaces2No wrap

1 nuevo archivo xD

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: devil013xl/Proyecto

base: master

head repository: Hustler013/Proyecto

compare: patch-1

✓ **Able to merge.** These branches can be automatically merged.

Create pull request

Discuss and review the changes in this comparison with others.

- 1 commit

± 1 file changed

0 commit comments

1 contributor

Commits on Jul 03, 2020

Hustler013

Create lol

Unverified9212306

Showing 1 changed file with 1 addition and 0 deletions.

UnifiedSplit

1 lol

@@ -0,0 +1 @@

1 + nuevo archivo xD

<> Code ⓘ Issues 🔗 Pull requests ⏮ Actions 📁 Projects 📖 Wiki 🛡 Security 📊 Insights

🔗 Hustler013:patch-1 had recent pushes less than a minute ago [Compare & pull request](#)

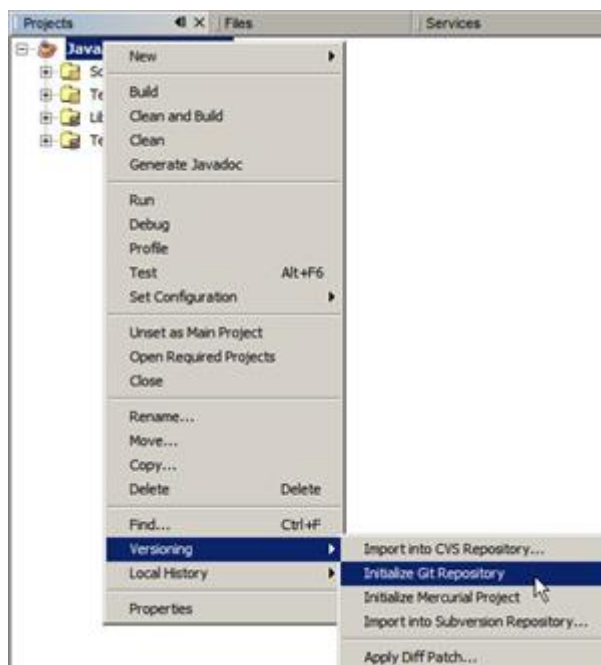
🔗 Branch: master [Go to file](#) [Add file](#) [Code](#)

👤 devil013xl committed 5d509ea 10 days ago ... ⌚ 3 commits 🔗 1 branch 🔖 0 tags

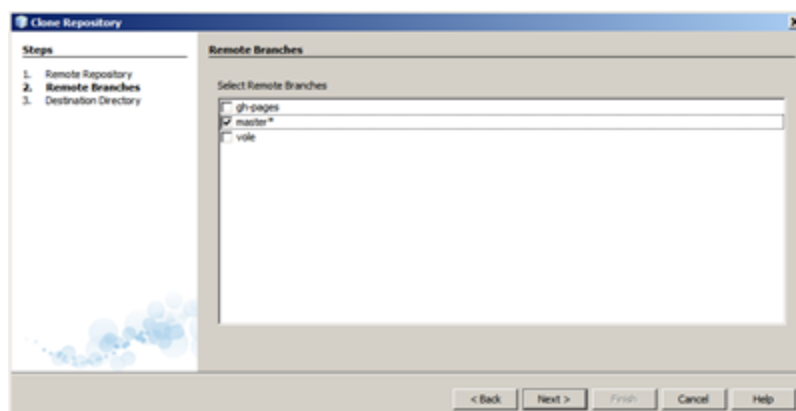
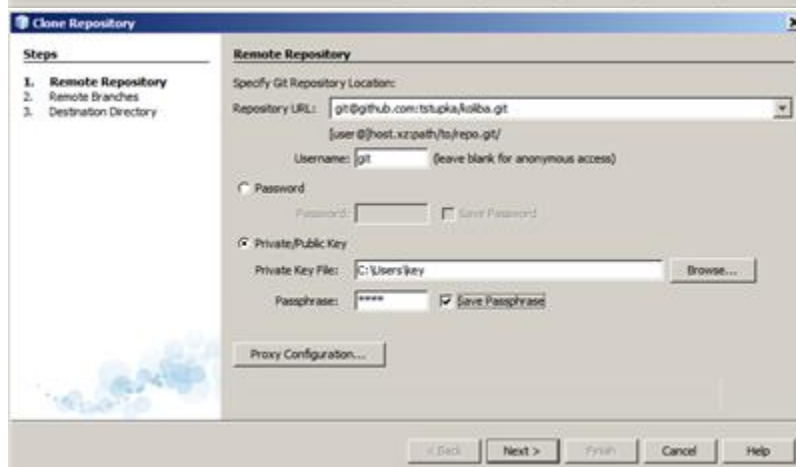
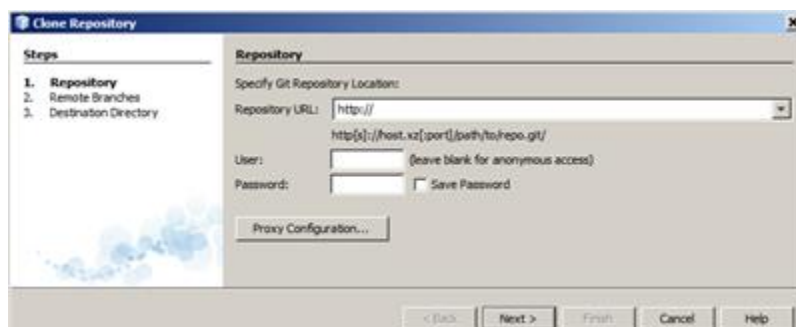
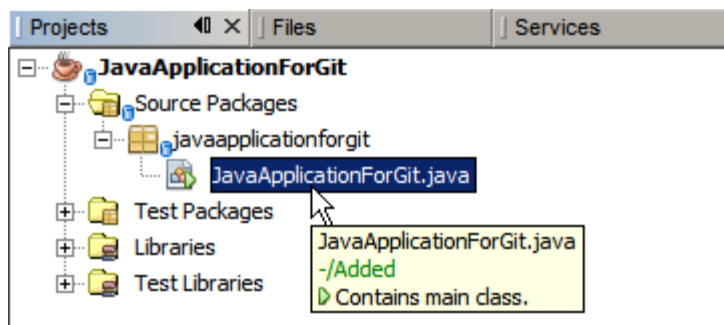
📄 README.md	Update README.md	12 months ago
📄 Screenshot_1.png	Add files via upload	10 days ago
📄 Screenshot_2.png	Add files via upload	10 days ago
📄 Screenshot_3.png	Add files via upload	10 days ago
📄 Screenshot_4.png	Add files via upload	10 days ago
📄 Screenshot_5.png	Add files via upload	10 days ago
📄 script.py	Add files via upload	10 days ago

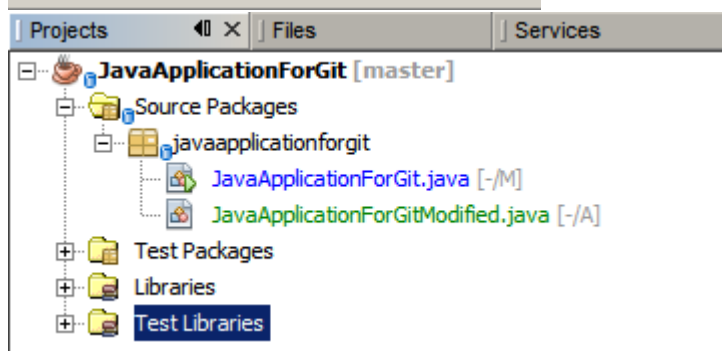
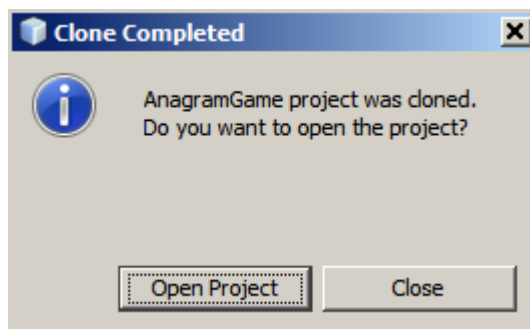
README.md

6. Investigar las herramientas de su IDE favorito para trabajar con Git y GitHub.



```
Output - GITRepository - C:\GITRepository
==[IDE]== 27.12.2010 18:44:18 Initializing ...
Initializing repository
Creating git C:\GITRepository\.git directory
git init C:\GITRepository
==[IDE]== 27.12.2010 18:44:18 Initializing ... finished.
```





Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).