

## 1. Resumen del documento:

### Objetivo principal:

*Creación y manejo de repositorios remotos usando Git y GitHub*

### Descripción general:

*Traslado de proyectos locales al controlador de versiones remoto GitHub dando énfasis a la creación del proyecto, edición y eliminación de archivos, creación de ramas, clonación de proyectos y otras herramientas básicas para la gestión de repositorios.*

## 2. Control de versiones:

Versión	Historia	Fecha	Responsable
1.0	Elaboración del documento	29/03/2023	Daniel Salvador Saavedra Quispe

## 3. Integrantes:

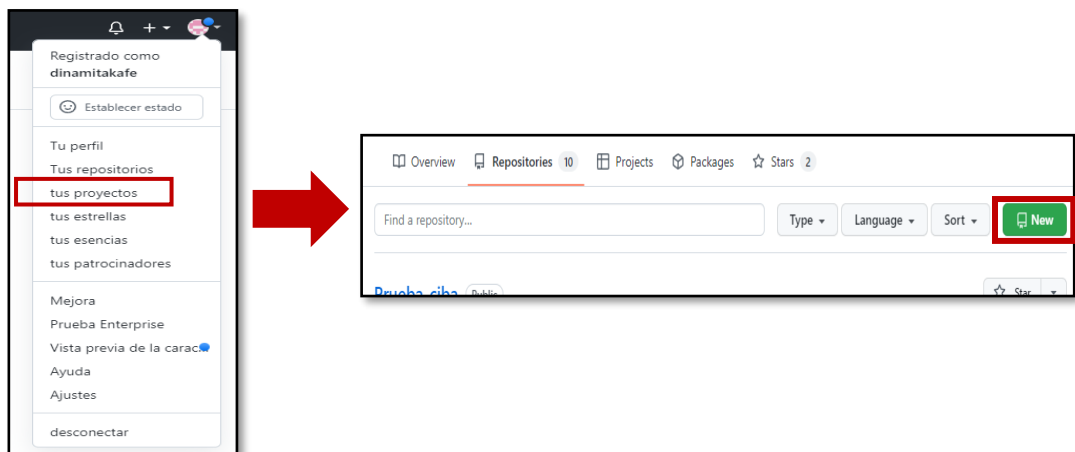
Nombres y Apellidos	Correo
Angie Villalobos Rodríguez	<a href="mailto:20180276@lamolina.edu.pe">20180276@lamolina.edu.pe</a>
Melannie Vasquez Peláez	<a href="mailto:20190315@lamolina.edu.pe">20190315@lamolina.edu.pe</a>
José Uribe Palomino	<a href="mailto:20190314@lamolina.edu.pe">20190314@lamolina.edu.pe</a>
Alejandro Salvador Cama	<a href="mailto:20190310@lamolina.edu.pe">20190310@lamolina.edu.pe</a>
Daniel Santini	<a href="mailto:20181184@lamolina.edu.pe">20181184@lamolina.edu.pe</a>
Efraín Huarínga	<a href="mailto:20181147@lamolina.edu.pe">20181147@lamolina.edu.pe</a>
Camila Briceño Francia	<a href="mailto:20180252@lamolina.edu.pe">20180252@lamolina.edu.pe</a>
Roger Roca Llamocca	<a href="mailto:20170188@lamolina.edu.pe">20170188@lamolina.edu.pe</a>
Caren Pampas Guadalupe	<a href="mailto:20160237@lamolina.edu.pe">20160237@lamolina.edu.pe</a>
Daniel Saavedra Quispe	<a href="mailto:20200336@lamolina.edu.pe">20200336@lamolina.edu.pe</a>

#### 4. Aspectos importantes para considerar:

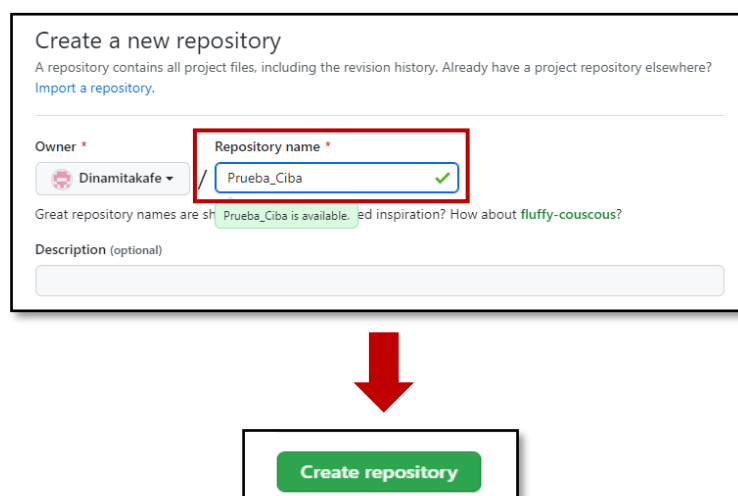
- Para hacer uso de un controlador de versiones de proyectos en remoto (GitHub) primero necesita tener instalado Git y poseer una cuenta en GitHub. Luego de tener el Git y el GitHub, ya podemos vincular ambos y crear nuestros repositorios en remoto.
  - Instalador de Git:  
<https://git-scm.com/>
  - Video de instalación de referencia de Git:  
<https://www.youtube.com/watch?v=EfroYRM1ZqE&list=LL&index=2>
  - Creación de cuenta en GitHub:  
[https://github.com/signup?ref\\_cta=Sign+up&ref\\_loc=header+logged+out&ref\\_page=%2F&source=header-home](https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home)

#### 5. Creando repositorios:

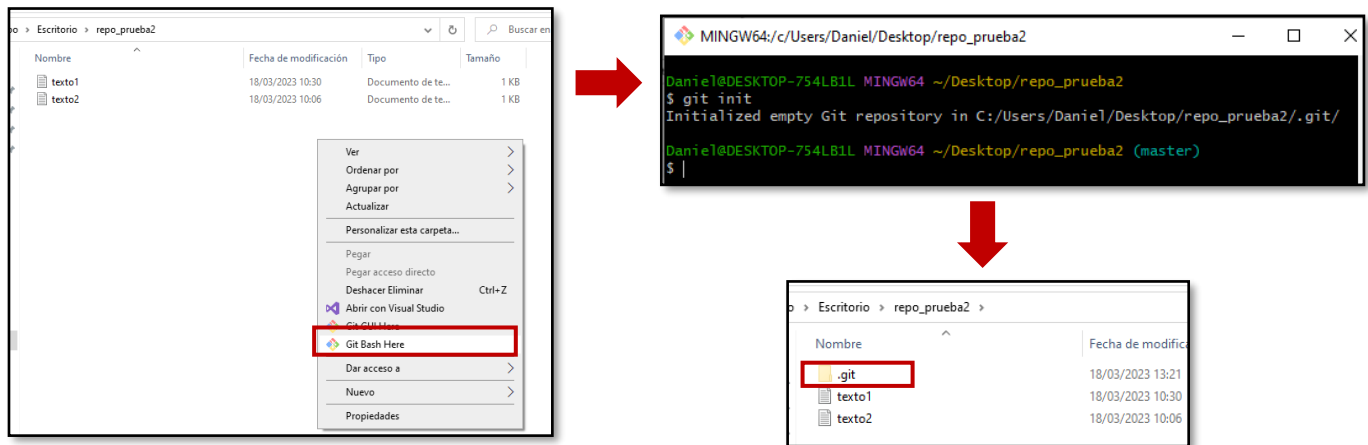
- Nos dirigimos a la sección <<Tus Repositorios>> y seleccionamos la opción <<Nuevo>>.



- Le damos un nombre a nuestro repositorio y seleccionamos <<Crear Repositorio>>, las demás opciones se pueden configurar luego.

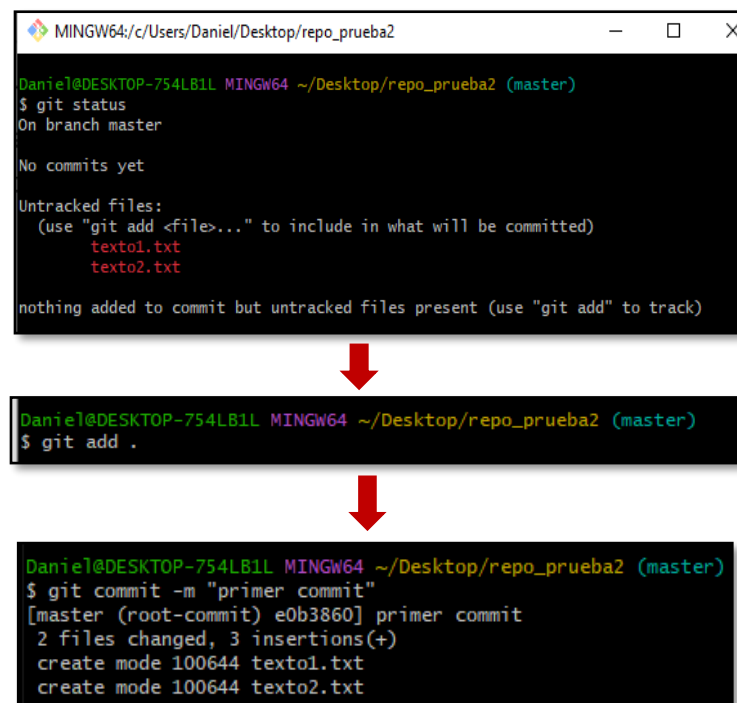


- Ahora nos vamos a la carpeta que se encuentra en nuestro local y deseamos subir, alojamos un Git Bash en dicha carpeta e iniciamos un repositorio local con el código `<< git init >>` y se nos creará una subcarpeta `<< .git >>` la cual indicará que el repositorio fue creado.



```
$ git init
```

- Subiremos nuestros archivos a nuestro repositorio local, usaremos el `<< git status >>` el cual nos dirá que archivos se encuentran fuera del repositorio o han sido editados, luego escribiremos `<< git add . >>` para agregar dichos archivos al repositorio y finalmente usaremos `<< git commit -m "..." >>`, esto guardará los cambios hechos y la versión en el repositorio.



```
$ git status
```

```
$ git add .
```

```
$ git commit -m "..."
```

- Ahora conectamos el GitHub con el Git, mediante el código que nos generó GitHub al momento de crear el repositorio, copiamos el código y lo pegamos en Git.

```
...or push an existing repository from the command line
git remote add origin https://github.com/Dinamitakafe/Prueba_Ciba.git
git branch -M main
git push -u origin main
```



```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (master)
$ git remote add origin https://github.com/Dinamitakafe/Prueba_Ciba.git

Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (master)
$ git branch -M main

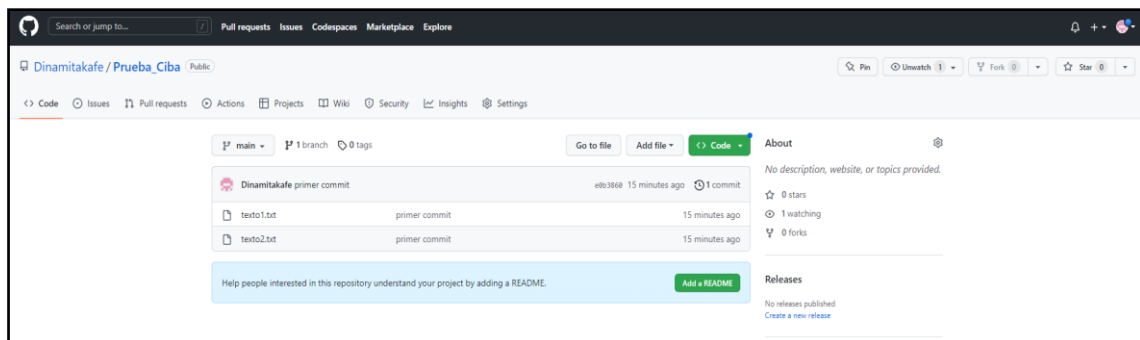
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 326 bytes | 326.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Dinamitakafe/Prueba_Ciba.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

```
$ git remote add origin URL
```

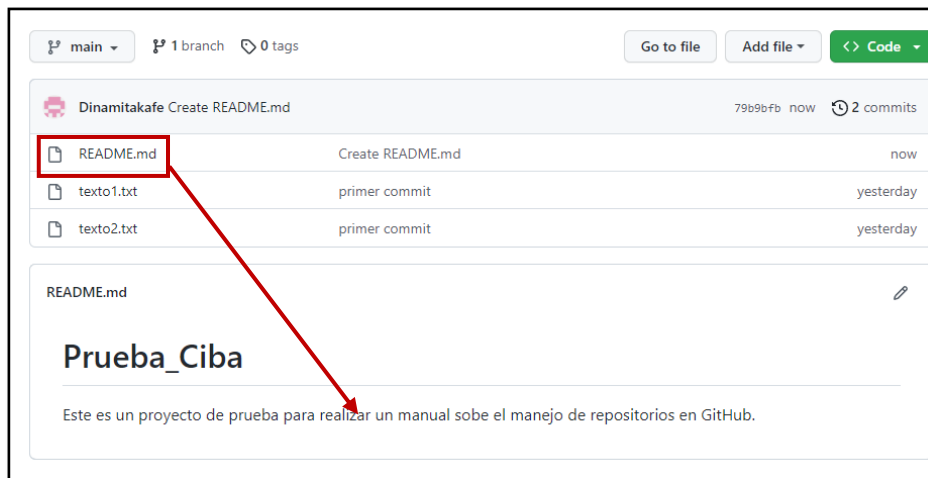
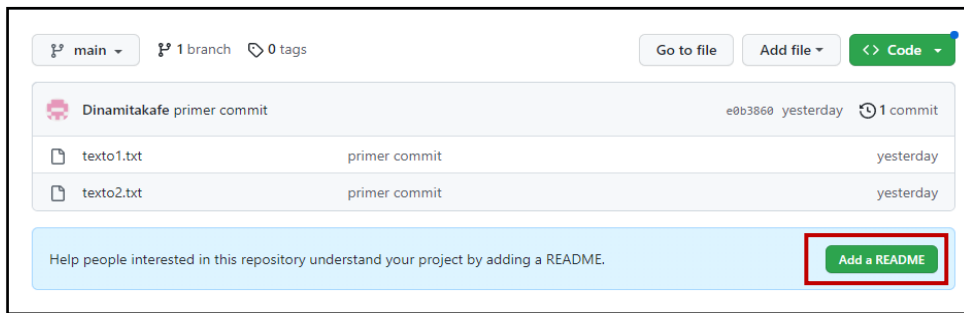
```
$ git branch -M main
```

```
$ git push -u origin main
```

- Verificamos y vemos que los archivos ya se subieron al repositorio remoto.



- Puedes agregar un <<README.md>> desde el remoto, dentro de este archivo puedes describir tu proyecto y se mostrara en la parte final de los archivos



## 6. Editando y añadiendo archivos en el repositorio:

- Luego de añadir o editar archivos (en este caso editamos el archivo texto2.txt y agregamos un nuevo archivo llamado WS\_Horizontal\_vertical\_MERCADOLIBRE.py), si queremos subirlos al repositorio en GitHub, lo que debemos hacer es subirlo primeramente al repositorio en local, en Git escribimos "git status" para ver los archivos que se encuentran fuera del repositorio.

```
MINGW64:/c/Users/Daniel/Desktop/repo_prueba2

Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   texto2.txt

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

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

Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ |
```

- Luego con un `<< git add . >>` los agregamos todos estos archivos al repositorio local y un `<< git status >>` para verificar que fueron agregados correctamente.

```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git add .

Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   WS_Horizontal_vertical_MERCADOLIBRE.py
    modified:   texto2.txt

Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ |
```

- Posteriormente agregamos un `<< git commit -m "... " >>` para describir los cambios que hicimos y se guarde la nueva versión.

```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git commit -m "Editando texto2 y añadiendo archivo .py"
[main 86db783] Editando texto2 y añadiendo archivo .py
 2 files changed, 69 insertions(+)
 create mode 100644 WS_Horizontal_vertical_MERCADOLIBRE.py

Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$
```

- Finalmente, con un `<< git push -u origin main >>` subiremos los cambios al repositorio remoto (GitHub).

```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git push -u origin main
To https://github.com/Dinamitakafe/Prueba_Ciba.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/Dinamitakafe/Prueba_Ciba.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

- Adicional: Si nos salta ese error hay que pullear el repositorio remoto al local primero, en Git escribimos << git pull >> para posteriormente escribir nuevamente << git push -u origin main >> y tendremos el repositorio remoto con las últimas modificaciones guardadas que realizamos.

```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 775 bytes | 129.00 KiB/s, done.
From https://github.com/Dinamitakafe/Prueba_Ciba
   e0b3860..79b9bfb  main      -> origin/main
Merge made by the 'ort' strategy.
 README.md | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 README.md
```



```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.04 KiB | 2.04 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Dinamitakafe/Prueba_Ciba.git
   79b9bfb..5808fa0  main -> main
branch 'main' set up to track 'origin/main'.
```



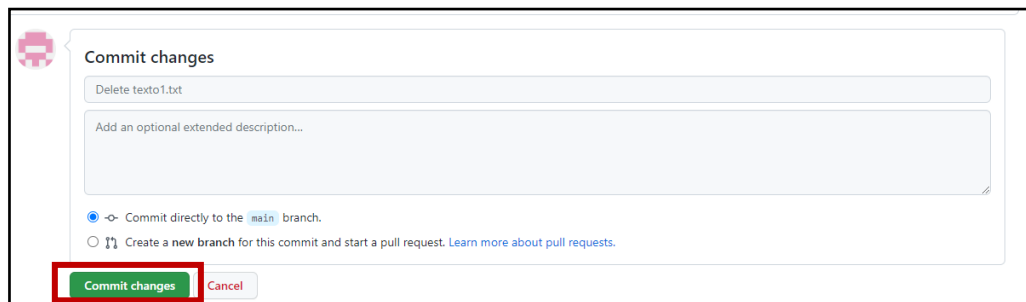
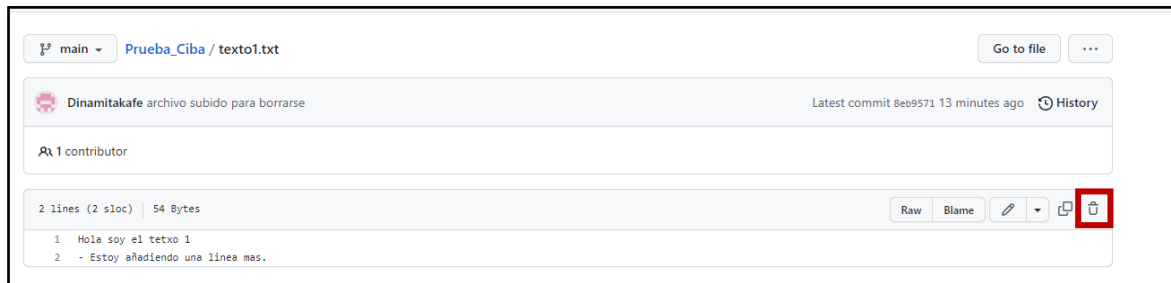
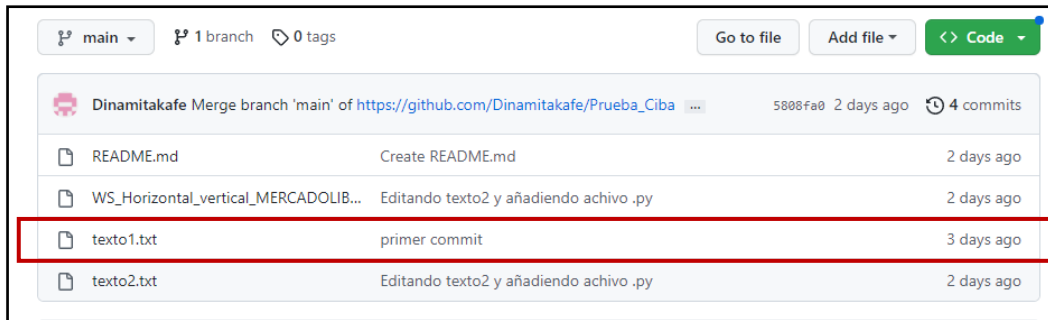
The screenshot shows the GitHub interface for the repository 'Dinamitakafe/Prueba\_Ciba'. It displays the commit history with the following details:

File	Commit Message	Time
README.md	Create README.md	37 minutes ago
WS_Horizontal_vertical_MERCADOLIB...	Editando texto2 y añadiendo archivo .py	13 minutes ago
texto1.txt	primer commit	yesterday
texto2.txt	Editando texto2 y añadiendo archivo .py	13 minutes ago

```
$ git status
$ git add .
$ git commit -m "...
$ git push -u origin main
$ git pull
$ git push -u origin main
```

## 7. Eliminando archivos en el repositorio:

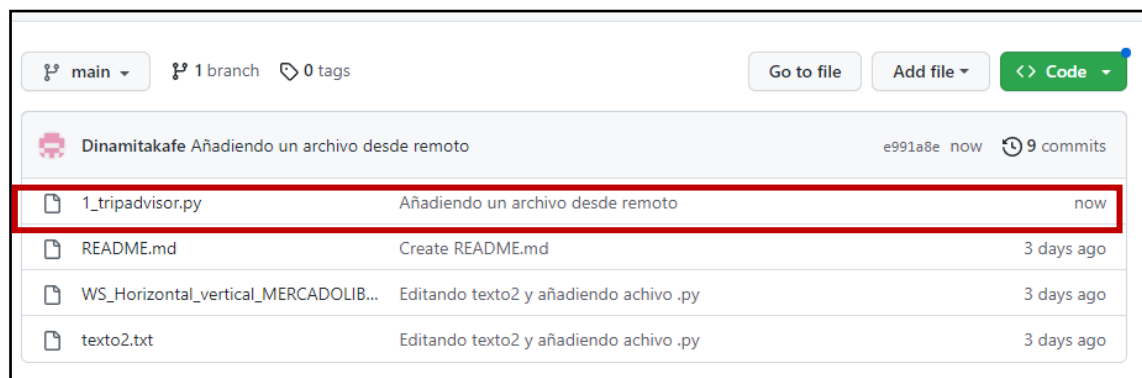
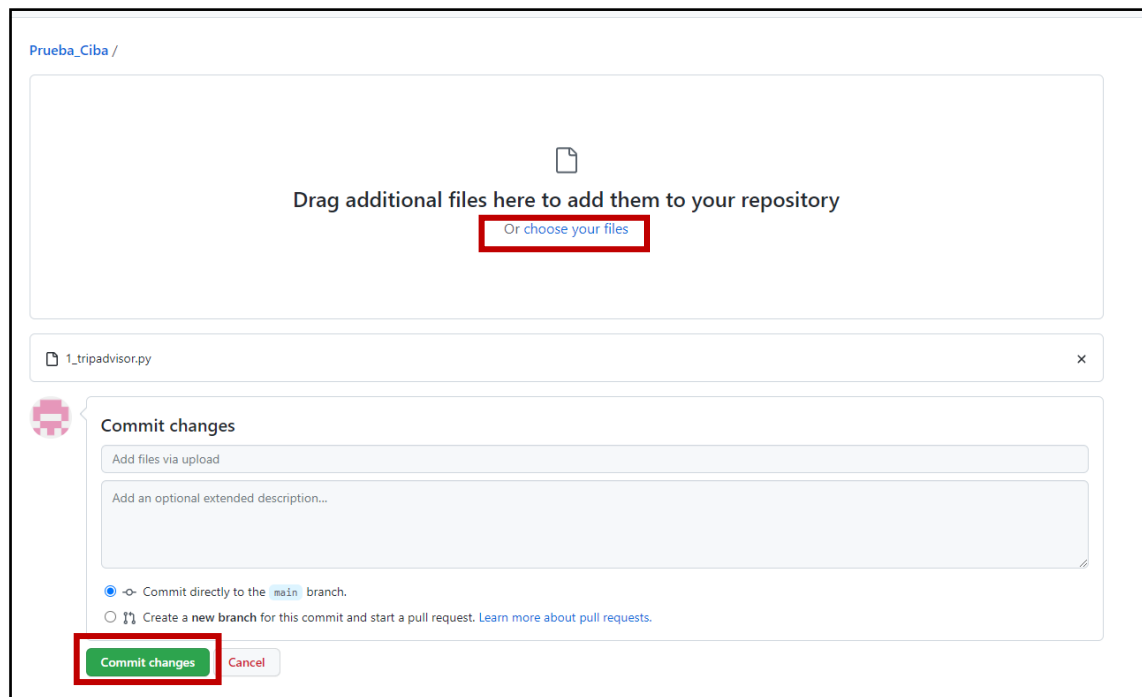
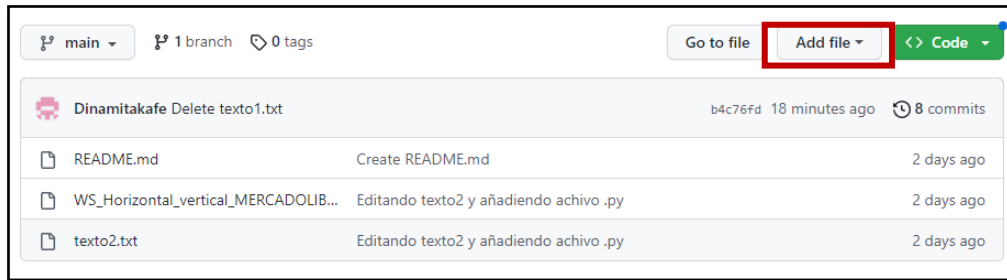
- Para eliminar un archivo, abrimos el archivo en GitHub y seleccionamos el icono de << tacho de basura >>.





## 8. Añadiendo archivos desde el remoto:

- Para subir un archivo desde GitHub tenemos que dirigirnos a la opción << Add file >>, luego escogemos el archivo y guardamos los cambios.



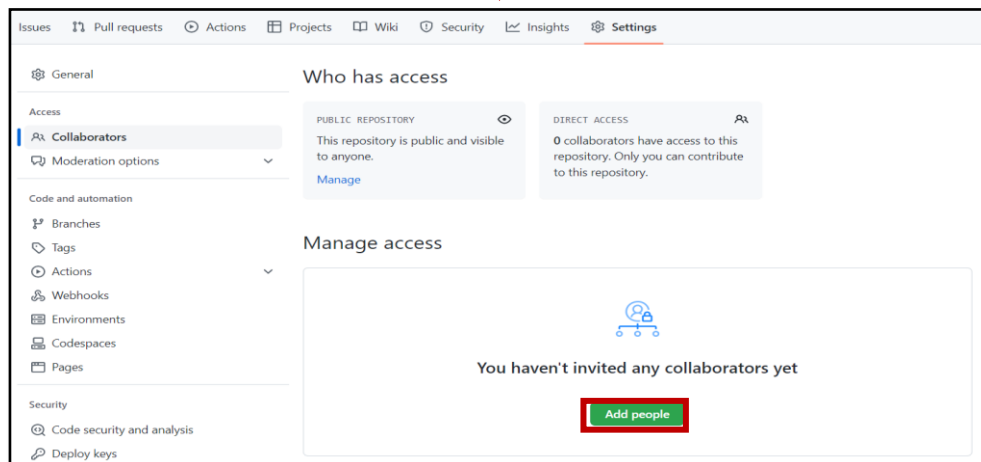
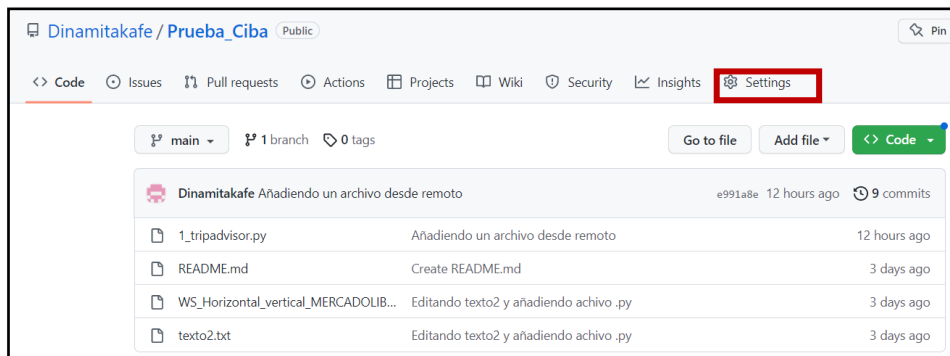
- Ahora con `<< git pull origin main >>`, actualizamos el repositorio local con los cambios hechos en el repositorio remoto.

```
Daniel@DESKTOP-7541R11 MINGW64 ~/Desktop/repo_prueba2 (main)
$ git pull origin main
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 2.77 KiB | 218.00 KiB/s, done.
From https://github.com/Dinamitakafe/Prueba_Ciba
* branch          main      -> FETCH_HEAD
   302877e..e991a8e  main      -> origin/main
Updating 302877e..e991a8e
Fast-forward
 1_tripadvisor.py | 71 +++++
 text01.txt       |  2 --
2 files changed, 71 insertions(+), 2 deletions(-)
create mode 100644 1_tripadvisor.py
delete mode 100644 text01.txt
```

\$ git pull origin main

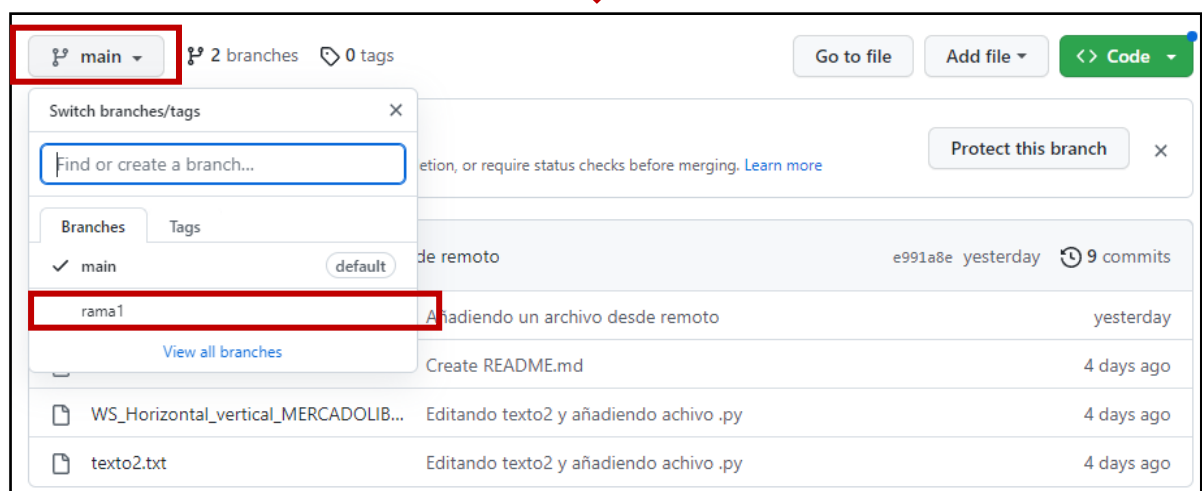
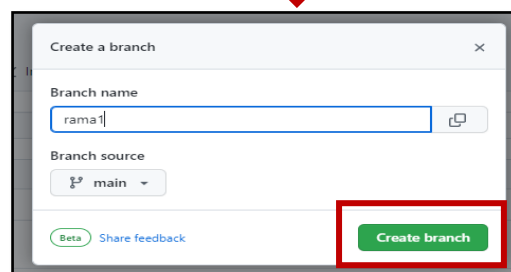
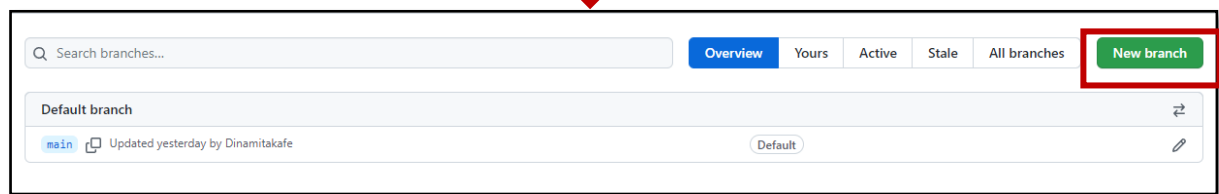
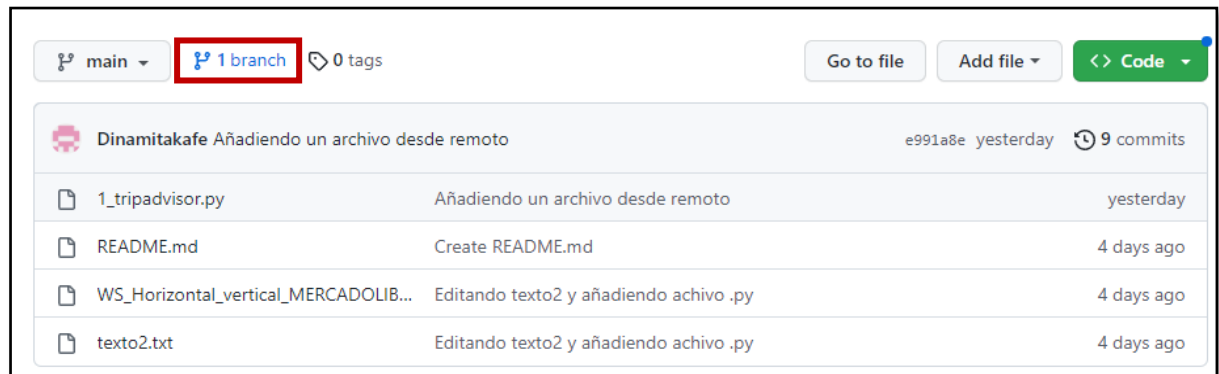
## 9. Añadiendo colaboradores a tu proyecto:

- Para añadir colaboradores a tu proyecto, es decir, otros usuarios que puedan editar archivos de tu repositorio en remoto, simplemente realizamos los siguientes pasos: “Settings (Ajustes)” >> “Collaborators” >> “Add people”. El invitado podrá ver los proyectos en tu cuenta, pero únicamente podrá editar en el proyecto en el cual se le otorgó el permiso.

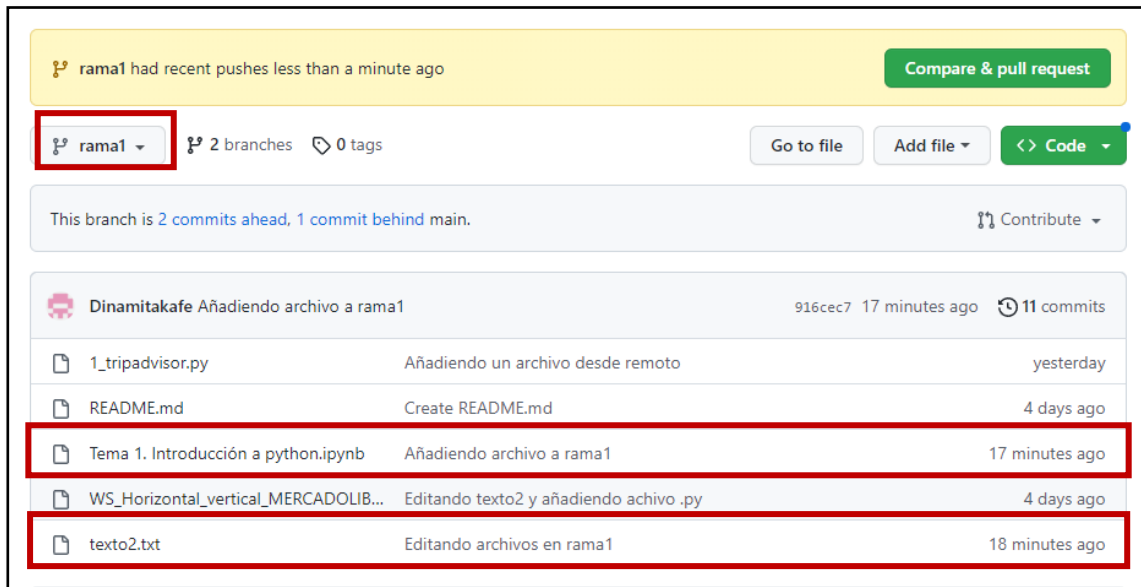


## 10. Creando ramas (Remoto):

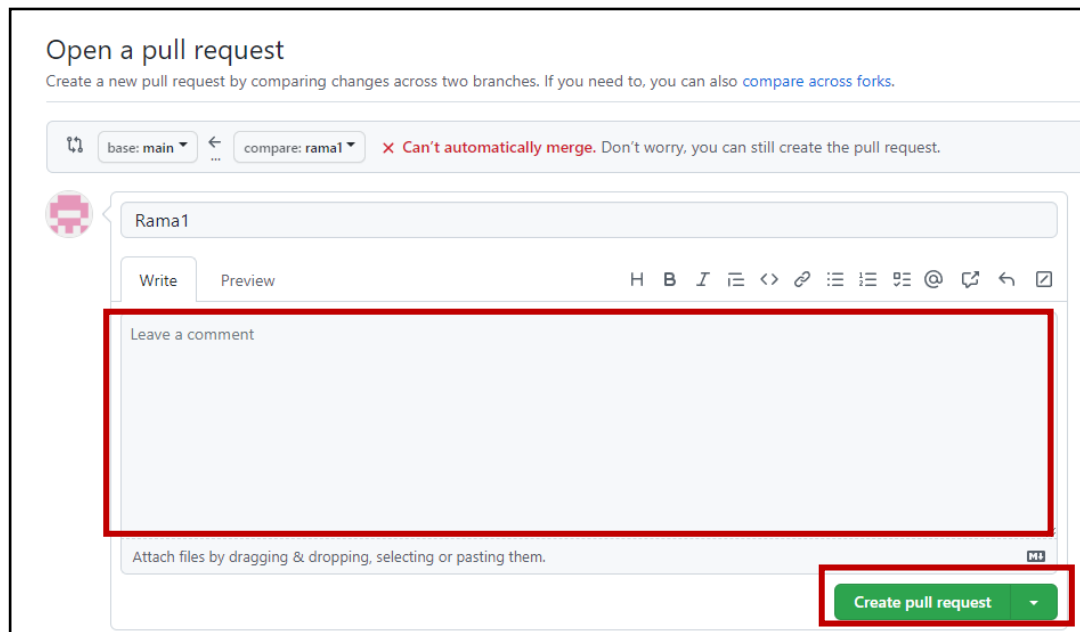
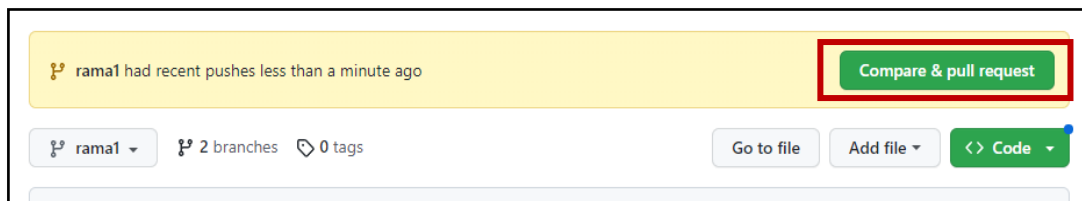
- Se usa una rama para identificar tareas de desarrollo sin afectar otras ramas en el repositorio. Cada repositorio tiene una rama por defecto y puede tener muchas otras ramas. Puedes fusionar una rama en otra rama usando una solicitud de extracción.
- Para crear una rama nueva, primero nos dirigimos a la opción << Branch >>, luego a la opción << New Branch >>, y le damos un nombre a nuestra rama, una vez concluido esos pasos verificamos que la rama haya sido creada exitosamente.

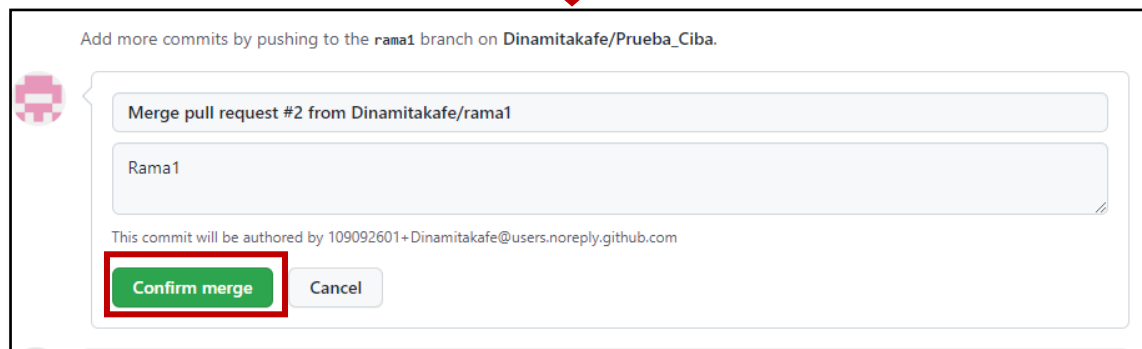
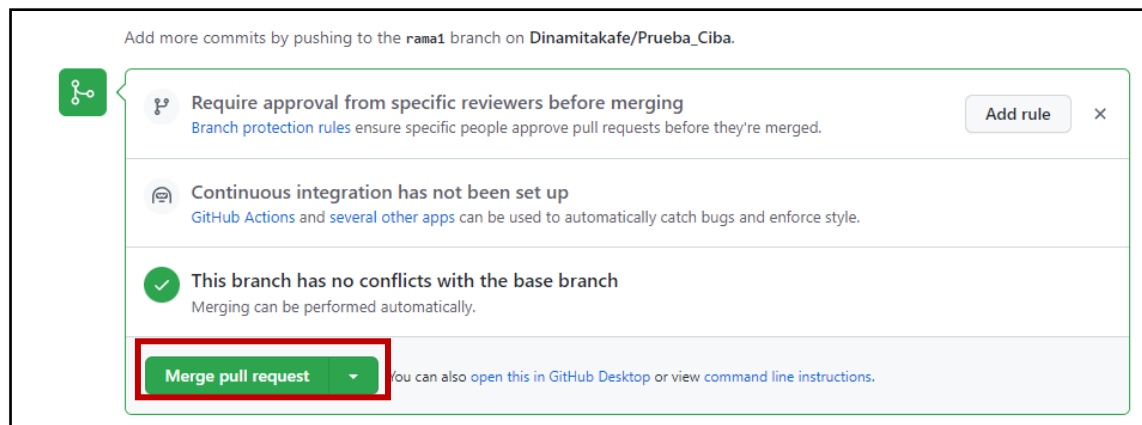


- Una vez creada la rama, podemos editar o añadir archivos desde el remoto sin alterar la rama original (main). En este caso, desde el remoto, editamos el archivo "texto2.txt" y añadimos un nuevo archivo << .ipynb >>.

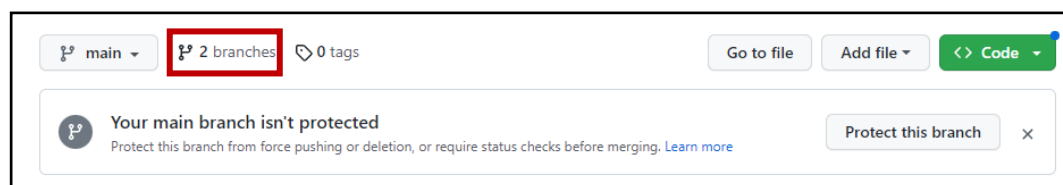


- Para fusionar las ramas, seleccionamos los siguientes pasos: "Compare & pull request" >> "Create pull request" (Puedes agregar un comentario) >> "Merge pull request" >> "Confirme merge".





- Para eliminar la rama pulleada y fusionada con la rama principal, nos dirigimos a la sección "Branch" y seleccionamos el icono de basurero, al eliminarlo nos quedamos con la rama principal guardada con el archivo que subimos en la rama1 (fusión).



## 11. Creando ramas (Local):

- Para crear una rama en el repositorio en local, abrimos Git en la carpeta donde se encuentra el repositorio y codificamos << git branch (nombre rama) >>.

```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git branch rama2
```

```
$ git branch nombrerama
```

- Luego de ello, para establecernos en dicha rama codificamos "git checkout (nombre rama)", realizamos los cambios añadiendo o editando archivos y verificamos que archivos están fuera del repositorio con un "git status".

```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ git checkout rama2
Switched to branch 'rama2'
```



```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (rama2)
$ git status
On branch rama2
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:   texto2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        scrapy-stackoverflow.ipynb

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

```
$ git checkout nombrerama
```

```
$ git status
```

- Adsd Agregamos los archivos con un << git add . >> para luego guardar y subir la nueva versión del repositorio con un << git commit -m "..." >>.

```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (rama2)
$ git add .
warning: in the working copy of 'scrapy-stackoverflow.ipynb', LF will be replaced by CRLF the next time Git touches it
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (rama2)
$ git commit -m "Añadiendo archivos desde la rama2"
[rama2 532107d] Añadiendo archivos desde la rama2
2 files changed, 178 insertions(+)
create mode 100644 scrapy-stackoverflow.ipynb
```

```
$ git add .
```

```
$ git commit -m "..."
```

- Por último, subimos todo lo editado en el local al remoto con `<< git push origin (nombre rama) >>`, luego de ello si revisamos nuestro repositorio en GitHub veremos que estará disponible la rama que creamos en Git.

```
Daniel@DESKTOP-7541B1L MINGW64 ~/Desktop/repo_prueba2 (rama2)
$ git push origin rama2
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 2.89 KiB | 2.89 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'rama2' on GitHub by visiting:
remote:   https://github.com/Dinamitakafe/Prueba_Ciba/pull/new/rama2
remote:
To https://github.com/Dinamitakafe/Prueba_Ciba.git
 * [new branch]      rama2 -> rama2
```



rama2 had recent pushes less than a minute ago [Compare & pull request](#)

rama2 2 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

Switch branches/tags

Find or create a branch...

Branches Tags

main default

✓ rama2 [View all branches](#)

la rama2	532107d 27 minutes ago	18 commits
Añadiendo un archivo desde remoto	2 days ago	
Create README.md	5 days ago	
Tema 1. Introducción a python.ipynb	Añadiendo archivo a rama1	yesterday
WS_Horizontal_vertical_MERCADOLIB...	Editando texto2 y añadiendo achivo .py	5 days ago
scrapy-stackoverflow.ipynb	Añadiendo archivos desde la rama2	27 minutes ago
texto2.txt	Añadiendo archivos desde la rama2	27 minutes ago

```
$ git push origin NombreRama
```

## 12. Clonación de carpetas en el local:

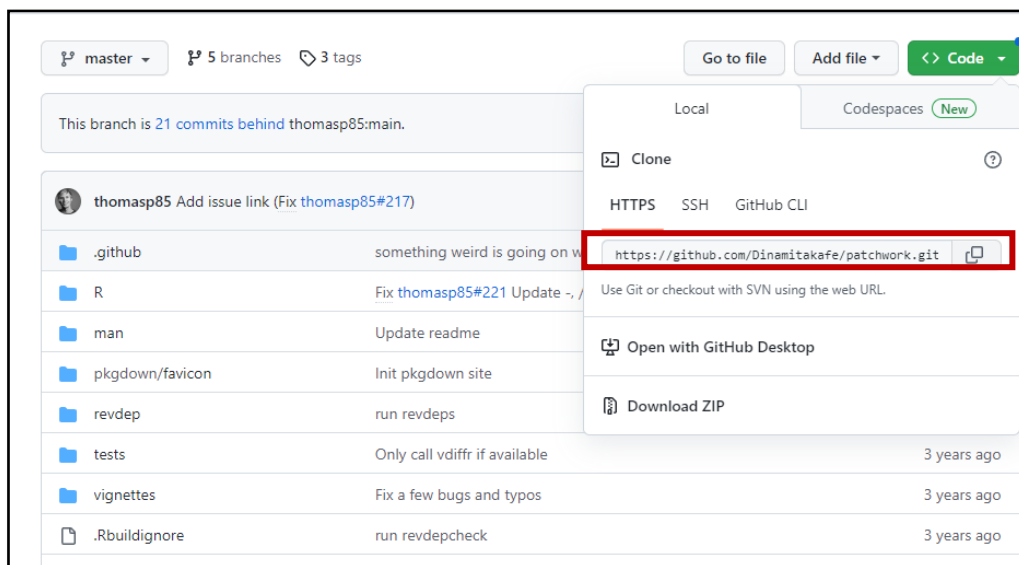
- En esta sección clonaremos carpetas enteras que podemos encontrar en GitHub y las colocaremos en nuestro escritorio local.
- El primer paso es salir de cualquier repositorio local en el que nos encontremos y salgamos al escritorio.

```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop/repo_prueba2 (main)
$ cd ..

Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop
$ |
```

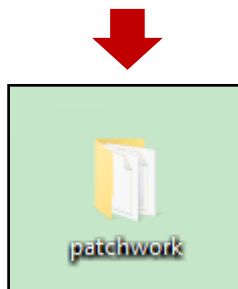
```
$ cd ..
```

- Escogemos la carpeta en GitHub que deseamos clonar y extraemos su URL.



- Ahora, usamos "git clone (url copiado)" en Git y tendremos la carpeta en nuestro escritorio local.

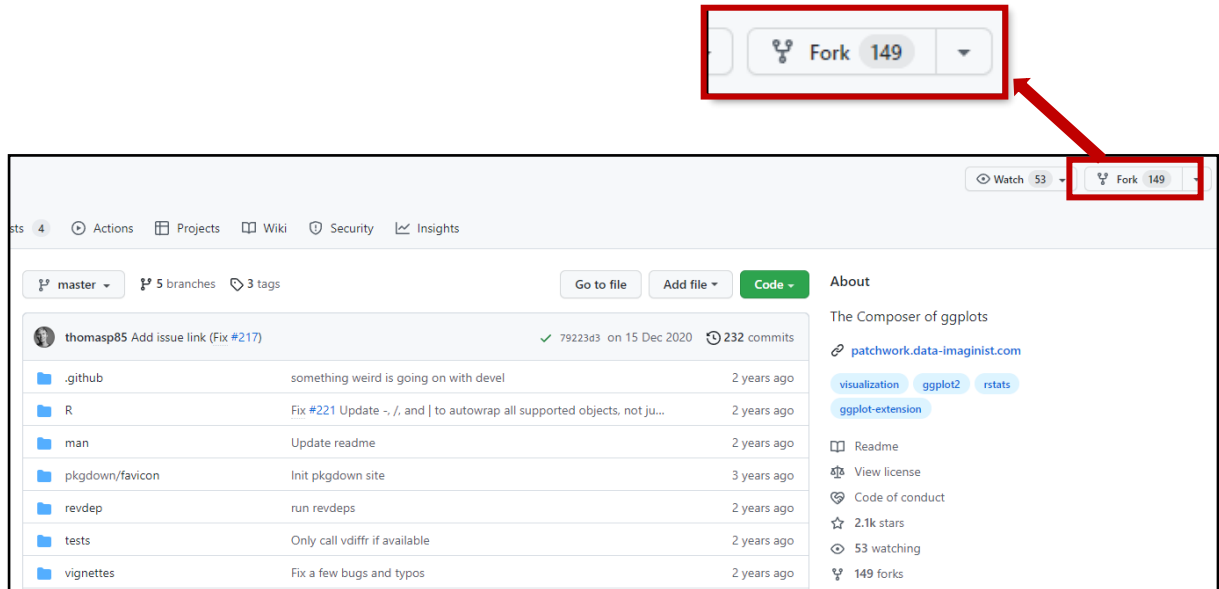
```
Daniel@DESKTOP-754LB1L MINGW64 ~/Desktop
$ git clone https://github.com/thomasp85/patchwork.git
Cloning into 'patchwork' ...
remote: Enumerating objects: 3413, done.
remote: Counting objects: 100% (1128/1128), done.
remote: Compressing objects: 100% (648/648), done.
remote: Total 3413 (delta 545), reused 1030 (delta 465), pack-reused 2285
Receiving objects: 100% (3413/3413), 36.76 MiB | 19.73 MiB/s, done.
Resolving deltas: 100% (1918/1918), done.
```





### 13. Clonación de carpetas en el remoto (forkear):

- En esta sección clonaremos carpetas enteras que podemos encontrar explorando GitHub y las colocaremos en nuestro espacio de repositorios propios en remoto, con esto podremos añadir y editar los archivos a nuestro antojo, sin alterar el repositorio original en el que se encuentra.
- Primero, escogemos el proyecto que deseamos clonar y nos dirigiremos a la opción “Fork”.



- Luego de ello, nos aparecerá una ventana el cual nos da la opción de cambiar o agregar un nombre a este proyecto clonado, procedemos a dar a la opción “Create Fork” y con esto tendremos el proyecto en la sección de nuestros repositorios.


### Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)


---

Owner \*

Repository name \*

 Dinamitakafe

 / 


patchwork 

By default, forks are named the same as their parent repository. You can customize the name to distinguish it further.

Description (optional)

The Composer of ggplots

---

 You are creating a fork in your personal account.

Create fork