



# Pasos para trabajar con Git (trabajando con repositorios).

Primeros pasos: Configuración

Creando nuestro primer repositorio.

Trabajando con un repositorio existente.

## Primeros pasos: Configuración

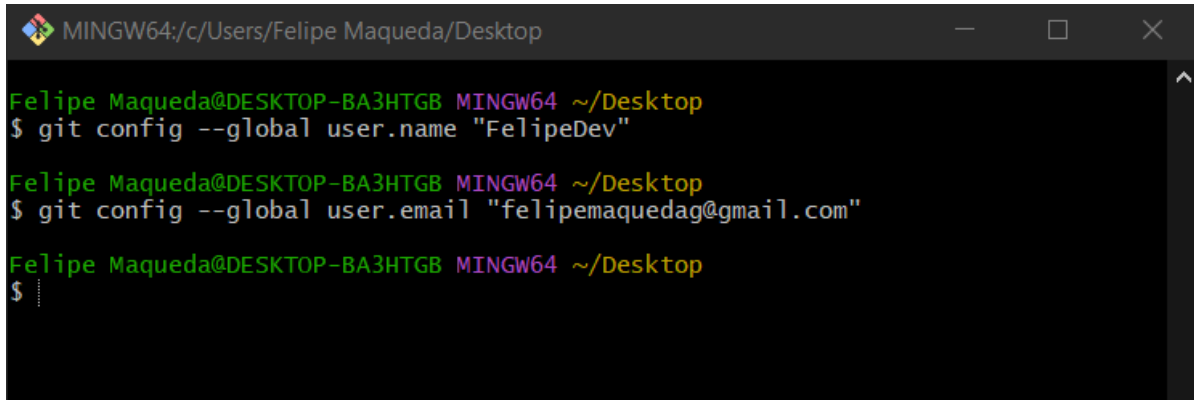
Tanto si es la **primera vez que trabajas con Git**, o has realizado un **cambio de equipo de cómputo** recientemente, es importante que sigas los siguientes pasos:

1. Valida que **Git se encuentre instalado** en tu computadora con el comando `git --version`.

```
MINGW64/c/Users/Felipe Maqueda/Desktop
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop
$ git --version
git version 2.41.0.windows.3
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop
$
```

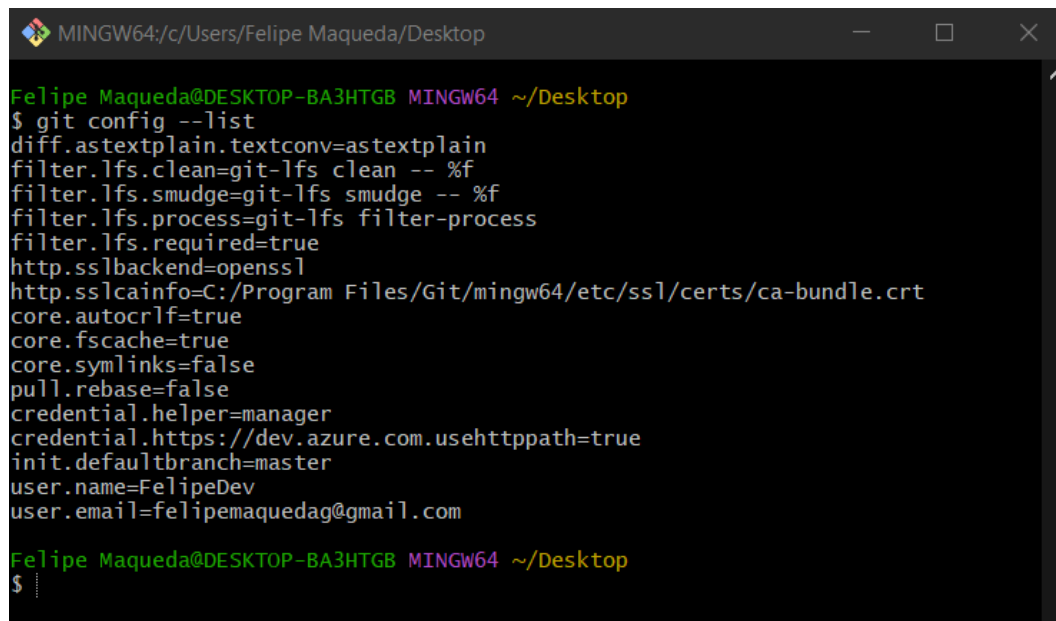
2. Ahora haremos la **configuración de nuestro nombre de usuario y correo electrónico** (el correo debe ser el que **utilizaste para realizar tu cuenta de GitHub**). Utilizaremos los siguientes comandos:

- a. `git config --global user.name "nombre de usuario"`
- b. `git config --global user.email felipemaquedag@gmail.com`



```
MINGW64/c/Users/Felipe Maqueda/Desktop
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop
$ git config --global user.name "FelipeDev"
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop
$ git config --global user.email "felipemaquedag@gmail.com"
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop
$
```

3. Si queremos **validar que nuestros datos hayan sido registrados correctamente**, utilizaremos el comando `git config --list`.

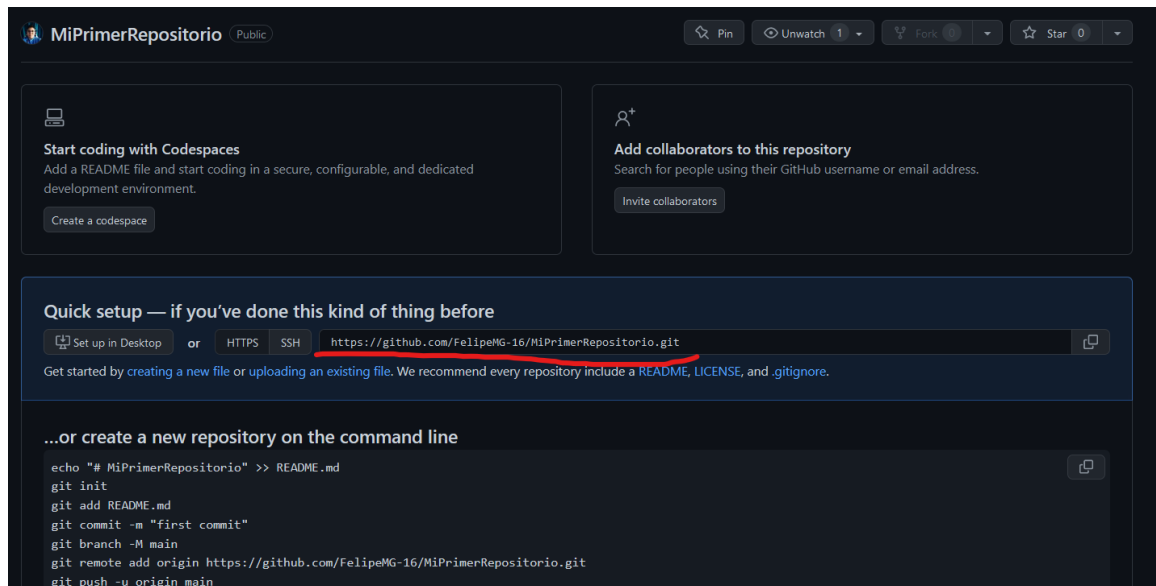


```
MINGW64/c/Users/Felipe Maqueda/Desktop
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=FelipeDev
user.email=felipemaquedag@gmail.com
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop
$
```

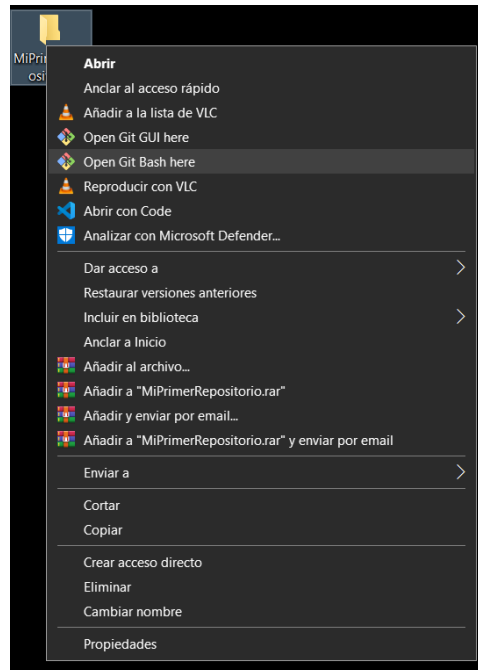
4. **Listo, ya tenemos todo listo para crear nuestro primer repositorio** (es importante mencionar que estos pasos **solo se realizan en una ocasión**).

# Creando nuestro primer repositorio.

1. Para **convertir una carpeta local en un repositorio de GitHub**, debemos crear un repositorio con el nombre deseado en nuestro perfil de GitHub (recuerda que el repositorio debe tener una visibilidad **“pública”**). Una vez que el repositorio esté hecho, **copia la dirección ya que le necesitaremos más adelante**.



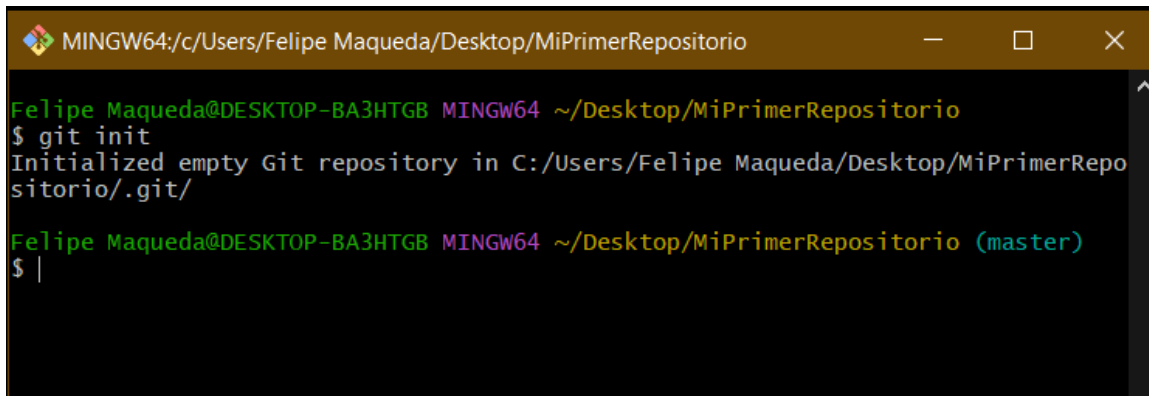
2. Ahora, en **nuestra computadora ubicaremos la carpeta que vamos a convertir en un repositorio**. Daremos clic derecho y seleccionaremos la opción **“Open Git Bash here”** para abrir nuestra **terminal de comandos posicionada en esa ruta**. También puedes moverte a la ruta deseada con el comando `cd RutaDeNuestraCarpeta/`.



3. Validamos en la terminal que nos encontremos en la ruta deseada.

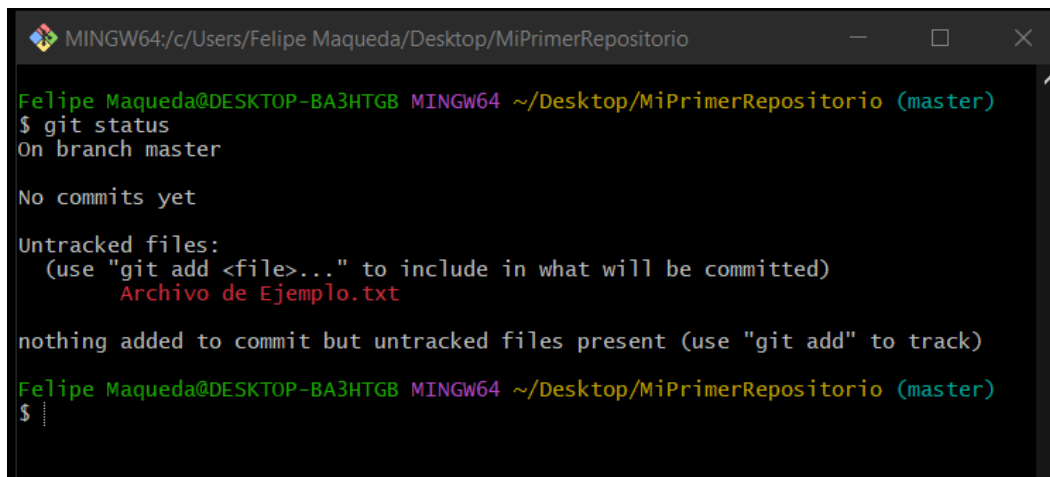


4. Comenzaremos **“convirtiendo”** nuestra carpeta local en un repositorio. Para esto utilizaremos el comando `git init`. Este comando nos permite **inicializar un repositorio en una carpeta local de nuestra computadora**. Nos daremos cuenta que se ha convertido en un repositorio porque **al final de la ruta aparece la palabra “master/main”**.



```
MINGW64/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio
$ git init
Initialized empty Git repository in C:/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio/.git/
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ |
```

5. El siguiente comando que utilizaremos es `git status`. Este comando nos **ayudará a mostrar el estado listando todos los elementos** que se encuentran “untracked”, es decir, **archivos que cuentan con alguna modificación pero que no se encuentran sincronizados con el repositorio en GitHub** (dichos archivos puede que se muestren en color rojo).



```
MINGW64/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Archivo de Ejemplo.txt

nothing added to commit but untracked files present (use "git add" to track)
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ |
```

6. Ahora debemos **agregar los archivos** que enviaremos al repositorio en GitHub. Para esto podemos utilizar el comando `git add .` para agregar **todos los archivos**, o `git add nombreArchivo.extensión` para agregar **uno por uno** los archivos pendientes de “trackear”. Una vez que hayamos ejecutado este comando, podemos utilizar nuevamente `git status` para validar que el documento se acaba de agregar al **staging area (zona de preparación)** y está **listo para agregarle un comentario** (se muestra de un color diferente).

```
MINGW64/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Archivo de Ejemplo.txt

nothing added to commit but untracked files present (use "git add" to track)
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git add .

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Archivo de Ejemplo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$
```

7. Para **agregar el comentario** utilizamos el comando `git commit -m` . Agregamos el parámetro `-m` ya que **de no hacerlo se abrirá el editor de texto por default** y tendremos que seguir las instrucciones necesarias para poder trabajar con el. **Agregando este parámetro podremos escribir el comentario en una sola linea de texto**. Es importante mencionar que el **comentario debe ir entre comillas dobles** `"` . Procura que el comentario incluya palabras que denoten **verbos o acciones**, y que sean lo mas específicos posible.

```
MINGW64/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git add .

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Archivo de Ejemplo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git commit -m "Se agrega archivo de ejemplo a mi primer repositorio"
[master (root-commit) 6fb76bb] Se agrega archivo de ejemplo a mi primer repositorio
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Archivo de Ejemplo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$
```

8. Antes de realizar el **push**, necesitamos **conectar la carpeta local con el repositorio de GitHub**. Para este paso, utilizaremos el comando `git remote add origin direccionDelRepositorio`. Recuerda **pegar la dirección del repositorio que copiamos anteriormente**. Si queremos **validar si la conexión se hizo correctamente**, podemos ejecutar el comando `git remote -v` y podremos **validar la conexión a nuestro repositorio en GitHub** (aparece **fetch** y **push** como opciones para el intercambio de información).

```
MINGW64/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Archivo de Ejemplo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git commit -m "Se agrega archivo de ejemplo a mi primer repositorio"
[master (root-commit) 6fb76bb] Se agrega archivo de ejemplo a mi primer repositorio
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Archivo de Ejemplo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git remote add origin https://github.com/FelipeMG-16/MiPrimerRepositorio.git

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git remote -v
origin https://github.com/FelipeMG-16/MiPrimerRepositorio.git (fetch)
origin https://github.com/FelipeMG-16/MiPrimerRepositorio.git (push)

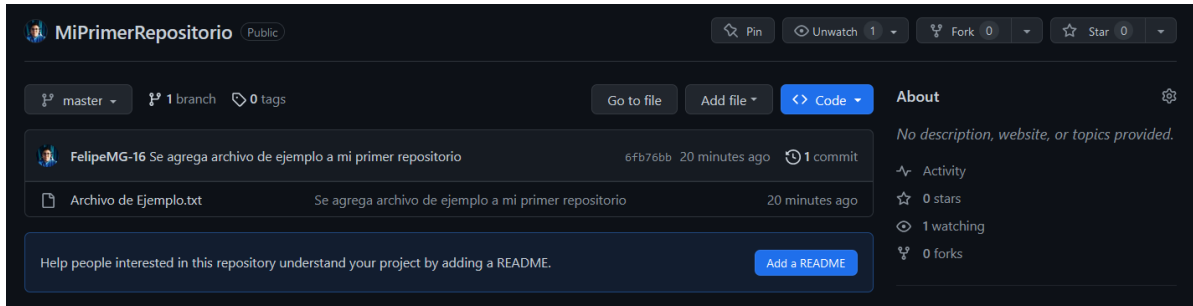
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$
```

9. Ya tenemos todo preparado para realizar el **push a nuestro repositorio**. Para este paso, utilizaremos el comando `git push origin master/main`. Es importante **validar el nombre de nuestra rama (aparece al final de nuestra ruta, puede ser master o main)**. Es posible que al hacer el push, se **nos solicite agregar un token** para validar la seguridad entre nuestra computadora y nuestra cuenta de GitHub (**para este paso revisa el manual correspondiente**).

```
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 255 bytes | 255.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FelipeMG-16/MiPrimerRepositorio.git
 * [new branch]      master -> master

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$
```

- Podemos **validar ahora en nuestra cuenta de GitHub** que el push se realizó **correctamente**. Podremos ver el **documento que creamos anteriormente**, con el comentario que agregamos con el `git commit -m`.



- !Felicidades!, lograste crear tu primer repositorio.**

## Trabajando con un repositorio existente.

Una vez que tenemos nuestro repositorio inicializado y ya conectado, **sólo necesitaremos “trackear” los nuevos cambios que vayan surgiendo (creación de nuevos documentos, modificación de archivos existentes, eliminación de archivos, etc.)**. Cuando esto sucede, mantendremos un **flujo de trabajo mucho mas corto**, es decir, ya **no necesitamos inicializar el repositorio o conectar el repositorio de GitHub**.

- Primero **validamos si hay cambios nuevos** utilizando el comando `git status`, y validamos los **archivos que cuentan con modificaciones**.



```
MINGW64:/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Nuevo archivo.txt

nothing added to commit but untracked files present (use "git add" to track)
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$
```

2. En este ejemplo, nos damos cuenta que **existe un nuevo archivo llamado “Nuevo Archivo.txt”** que necesita ser **“trackeado”**, es decir, **agregarlo al staging area para poder agregarle un comentario**. Recuerda utilizar `git status` para **validar que el archivo se ha agregado correctamente**.

```
MINGW64:/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Nuevo archivo.txt

nothing added to commit but untracked files present (use "git add" to track)
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git add Nuevo\ archivo.txt
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Nuevo archivo.txt
Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ |
```

3. Continuamos con el flujo de trabajo que utilizamos anteriormente. Para este caso, utilizamos `git commit -m` para **agregar nuestro comentario**. Recuerda **mantener las buenas prácticas al realizar esta acción**.

```
MINGW64:/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio
(use "git add <file>..." to include in what will be committed)
Nuevo archivo.txt

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

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git add Nuevo\ archivo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Nuevo archivo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git commit -m "Se agrega nuevo archivo al repositorio original"
[master 6908ae9] Se agrega nuevo archivo al repositorio original
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Nuevo archivo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$
```

4. Por último, utilizamos el comando `git push origin master` para **subir estos nuevos cambios al repositorio de GitHub**.

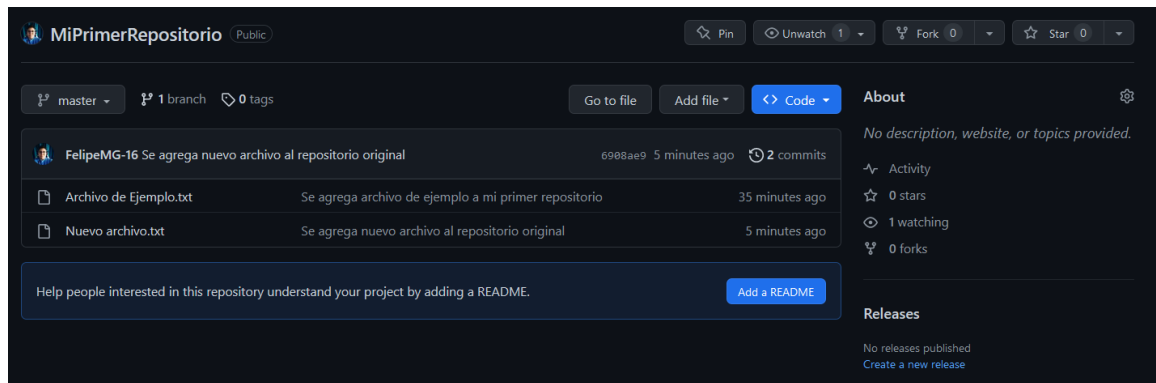
```
MINGW64:/c/Users/Felipe Maqueda/Desktop/MiPrimerRepositorio
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Nuevo archivo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git commit -m "Se agrega nuevo archivo al repositorio original"
[master 6908ae9] Se agrega nuevo archivo al repositorio original
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Nuevo archivo.txt

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 288 bytes | 288.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FelipeMG-16/MiPrimerRepositorio.git
6fb76bb..6908ae9 master -> master

Felipe Maqueda@DESKTOP-BA3HTGB MINGW64 ~/Desktop/MiPrimerRepositorio (master)
$
```

5. Podemos validar que **los cambios ya se encuentran reflejados** en el repositorio de GitHub.



## 6. ¡Listo, ya puedes trabajar de forma continua con un repositorio existente!