



Primeros pasos y Git Init

1. En el buscador de Windows ingresar: **git bash**
2. Una vez abierta la terminal, escribir **pwd** lo cual nos va a permitir saber en que parte del directorio estamos. Viene del inglés "present working directory"
3. Luego, vamos a ir a la carpeta que deseemos. En este caso **Git_prueba**, que se encuentra en **Desktop** (escritorio). Para esto vamos a usar el comando **cd** (de "change directory"), tal como se ve en la imagen.
4. Por ultimo, vamos a inicializar git, ingresando: **git init**
5. Opcional: ¡**pwd** se puede corroborar en cada paso que vamos realizando!

```
MINGW64:/c/Users/frann/Desktop/Git_prueba

frann@DESKTOP-IUUDUOI MINGW64 ~ (main)
$ pwd
/c/Users/frann

frann@DESKTOP-IUUDUOI MINGW64 ~ (main)
$ cd Desktop/Git_prueba

frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (main)
$ git init
Initialized empty Git repository in C:/Users/frann/Desktop/Git_prueba/.git/

frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$ pwd
/c/Users/frann/Desktop/Git_prueba

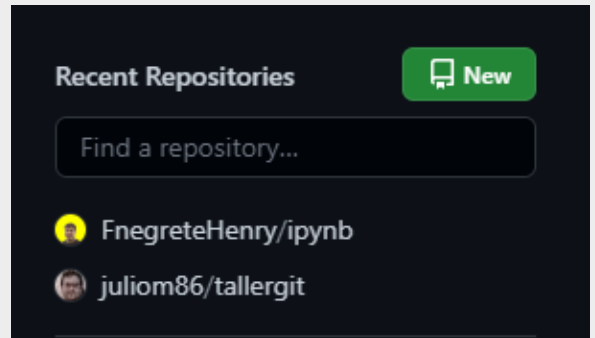
frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$
```

Como crear un Repo en Github

1. En el inicio de **Github**, clickear en **New**.

Pueden ver dos repositorios creados por los
HM's. O también desde

<https://github.com/new>

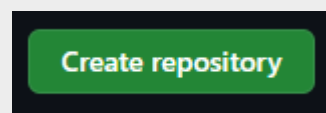


2. Creamos un nuevo repositorio con el nombre deseado.



Para este caso, el repo va a ser **Público**.

Además, también podemos agregar un archivo **README**.

3. Por último, le damos al botón de **Create repository**.




Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/FnegreteHenry/git_henry.git` 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


...or create a new repository on the command line

```
echo "# git_henry" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/FnegreteHenry/git_henry.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/FnegreteHenry/git_henry.git
git branch -M main
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

1. Los pasos anteriores nos van a llevar a un **Quick Setup**. La **url** que se generó, nos va a permitir comunicarnos desde el repositorio local con Github.

```
C:\> MINGW64:/c:/Users/frann/Desktop/Git_prueba

frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$ git config user.name "FnegreteHenry"

frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$ git config user.email "fnegrete@soyhenry.com"

frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$ git remote add origin "https://github.com/FnegreteHenry/git_henry.git"

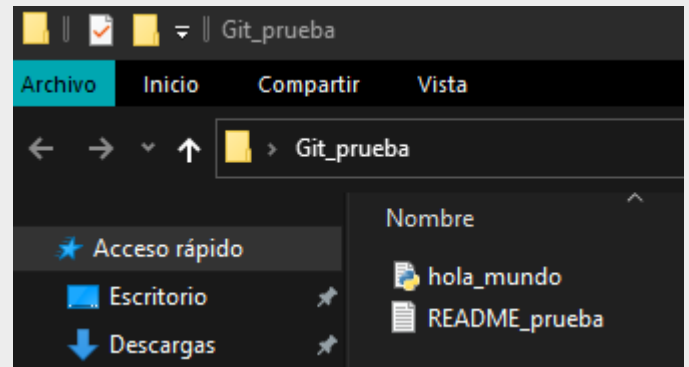
frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$
```

2. Con **git config user.name** y **git config user.email** vamos a introducir nuestros datos para poder conectarnos con el repositorio remoto.
3. Para comunicar los archivos locales con la repo remota usamos **git remote add origin** y entre comillas, la url que podemos ver en la imagen de arriba.



Crear un archivo local y subirlo al repositorio remoto

1. Creamos una nueva carpeta y ponemos archivos de prueba, por ejemplo, un **.txt** y un **.py**.



2. Con **git status** comprobamos el estado de los nuevos archivos.

Con **git add .**

añadimos todos los archivos que tienen cambios.

Poniendo otra vez

git status

comprobamos que los cambios se hayan hecho.

Para confirmar los cambios, vamos a

usar **git commit -m**

y el nombre del commit entre comillas.

```
frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$ git status
On branch master

No commits yet

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

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

frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$ git add .

frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$ git status
On branch master

No commits yet

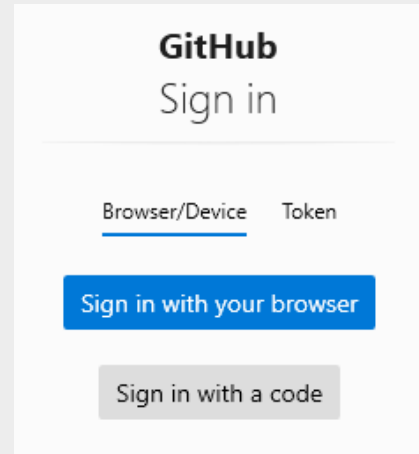
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README_prueba.txt
        new file:   hola_mundo.py

frann@DESKTOP-IUUDUOI MINGW64 ~/Desktop/Git_prueba (master)
$ git commit -m "primer commit"
[master (root-commit) 8b3fd51] primer commit
2 files changed, 1 insertion(+)
create mode 100644 README_prueba.txt
create mode 100644 hola_mundo.py
```

3. Con **git push -u origin master** vamos a hacer la conexión con Github. Master o main son la rama principal con la que vamos a trabajar. Origin es el nombre por convención que le ponemos a nuestro repositorio.

```
frann@DESKTOP-IUUDU0I MINGW64 ~/Desktop/Git_prueba (master)
$ git push -u origin master
```

5. Si es la primera vez que **pusheamos** desde nuestra compu, Github nos va a pedir validar usuario y contraseña. Por lo que vamos a clicar **Sign in with your browser**.



4. Luego de clicar el botón azul, se va a abrir en nuestro navegador una nueva ventana, donde nos va a aparecer lo mismo que en la imagen de la izquierda. Simplemente, le damos a **Authorize GitCredentialManager**.
6. Por último, entramos de nuevo a nuestro repositorio creado en Github, y vamos a ver que se subieron los archivos que habíamos creado desde nuestra carpeta local. **¡Ahora otros usuarios van a poder ver nuestro repo!**

