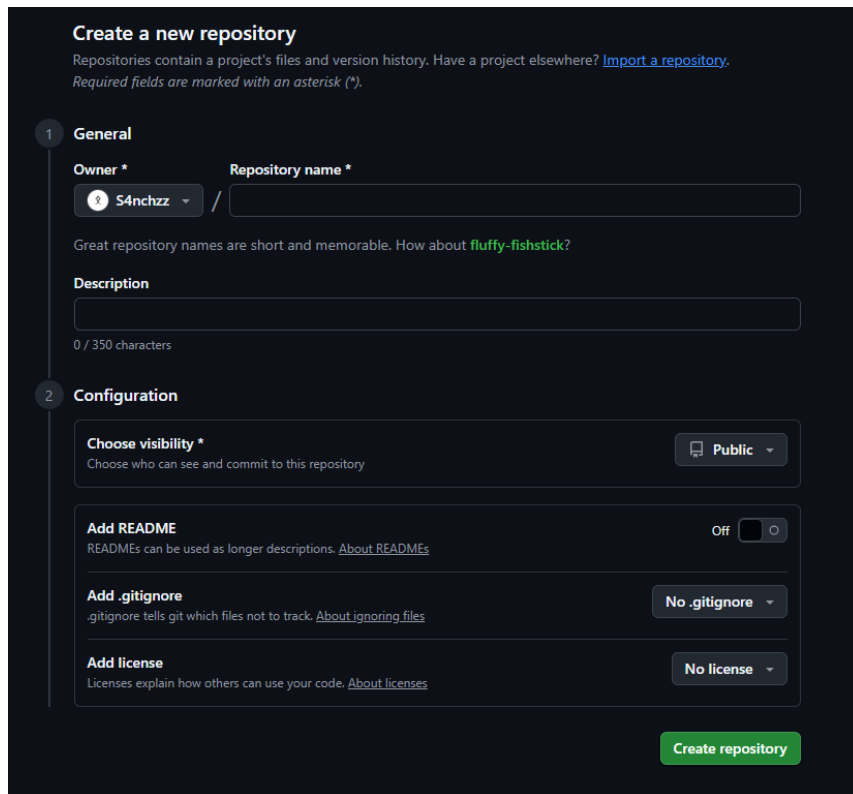


# Manual de usuario

Vinculación VsCode con github.

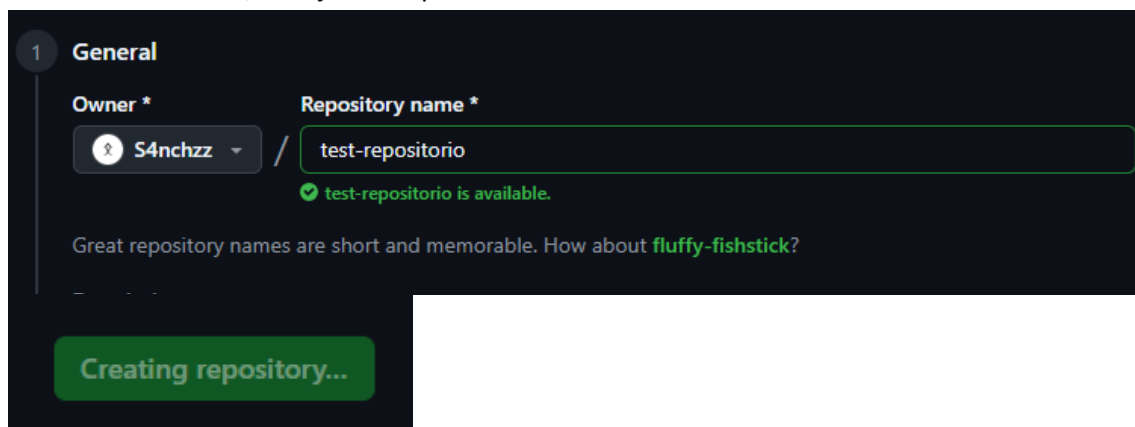
## Creación de repositorios

1. Para comenzar, crearemos un repositorio en nuestro Github de forma remota, para ello accederemos a <https://github.com/new> con nuestra sesión iniciada de antemano.



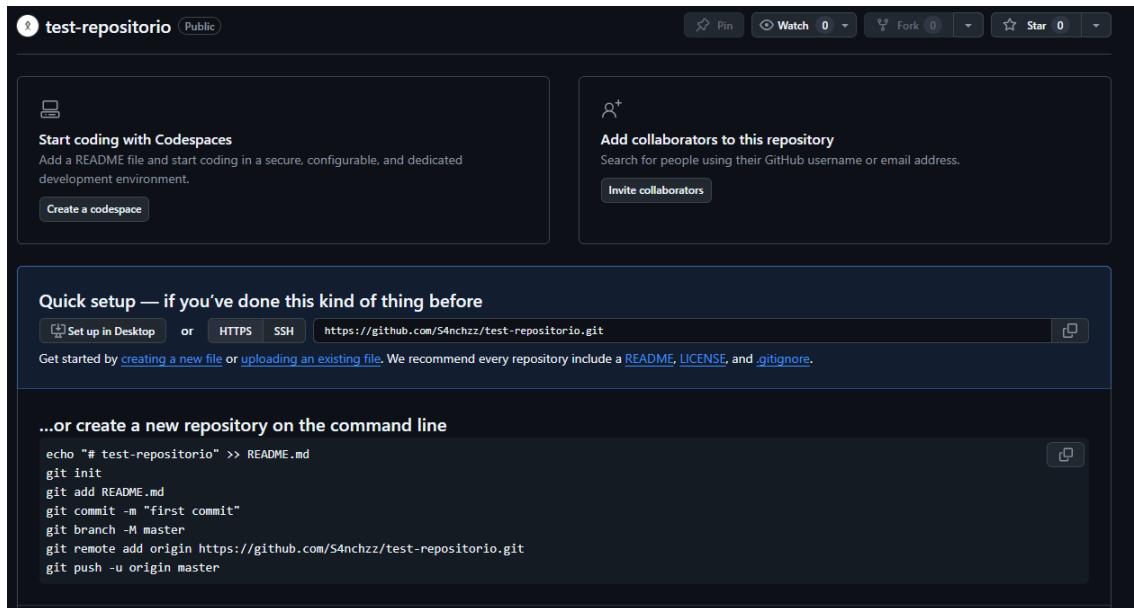
The screenshot shows the 'Create a new repository' page on GitHub. It is divided into two sections: 'General' and 'Configuration'. In the 'General' section, the 'Owner' is set to 'S4nchzz' and the 'Repository name' field is empty. A suggestion 'fluffy-fishstick?' is shown. The 'Description' field is also empty. In the 'Configuration' section, 'Choose visibility' is set to 'Public', 'Add README' is turned off, 'Add .gitignore' is set to 'No .gitignore', and 'Add license' is set to 'No license'. A green 'Create repository' button is at the bottom right.

2. A continuación, elegiremos un nombre el cual tendrá el repositorio, si es privado o no, si queremos añadir un README, si queremos añadir un gitignore o si queremos añadir una licencia, lo dejaremos por defecto.



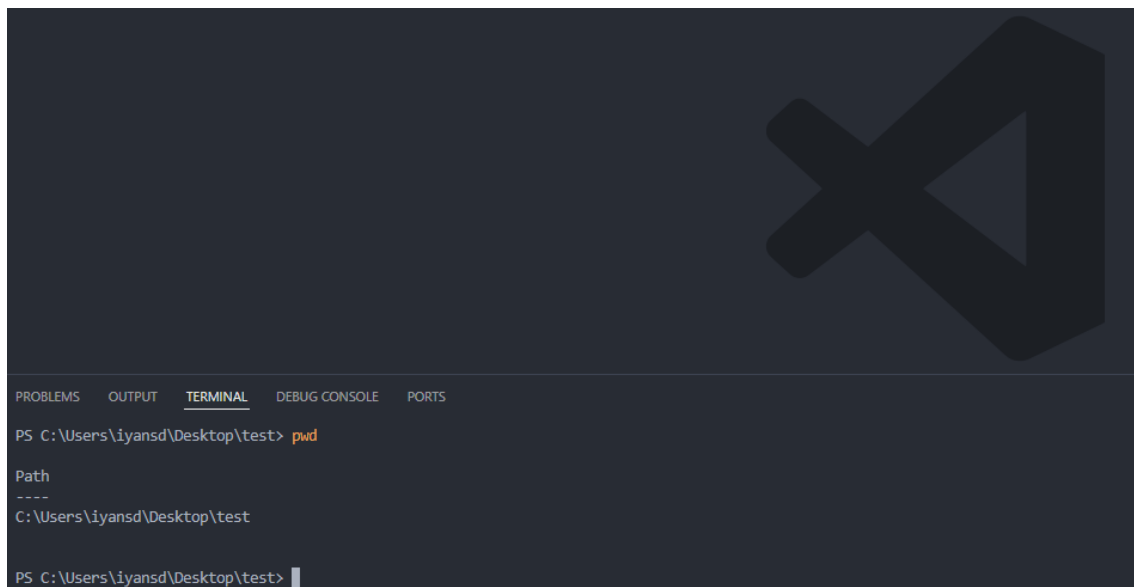
This screenshot shows the same 'Create a new repository' form, but now the 'Repository name' field contains 'test-repositorio'. A green checkmark and the message 'test-repositorio is available.' are displayed below the field. The 'Creating repository...' button is now visible at the bottom left.

- Una vez creado el repositorio nos aparecerá la siguiente ventana, esto querra decir que el repositorio remoto en github se ha creado sin problemas y nos da una guía para vincular el repositorio remoto con el local para poder empezar a añadir contenido.



## Vinculación del repositorio local con el repositorio remoto.

- Para vincular el repositorio local con el remoto primeramente crearemos una carpeta en nuestro equipo y abriremos esta con VsCode



2. A continuación, abriremos una terminal (en mi caso git bash para usar comandos de linux) en este directorio en la cual pondremos todos los comandos necesarios para crear y vincular el repositorio local, el primer comando que pondremos será *git init* para inicializar el repositorio local.

```
iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test
$ git init
Initialized empty Git repository in C:/Users/iyansd/Desktop/test/.git/

iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
$ ls -la
total 8
drwxr-xr-x 1 iyansd 1049089 0 Sep 29 10:44 ./
drwxr-xr-x 1 iyansd 1049089 0 Sep 29 10:35 ../
drwxr-xr-x 1 iyansd 1049089 0 Sep 29 10:44 .git/

iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
$
```

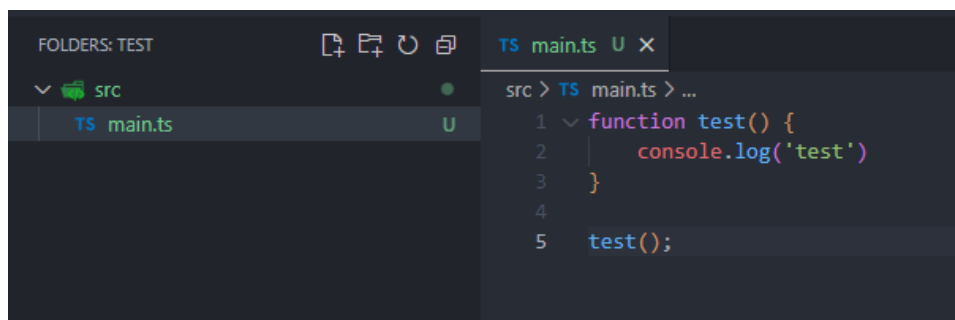
Como podemos ver en la imagen anterior se ha creado .git, esta carpeta contendrá todos los commits HEADS branches y demás datos que github usara para su control de flujo

3. Ahora crearemos la rama local la cual contendrá el mismo nombre que la **rama por defecto** de nuestra cuenta de github

```
iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
$ git branch -M master

iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
$
```

4. Ahora vamos a crear un archivo el cual lo añadiremos al repositorio **local** para subirlo al repositorio **remoto** más adelante.



```
FOLDERS: TEST
src
  TS main.ts

TS main.ts U x
src > TS main.ts > ...
1  function test() {
2      console.log('test')
3  }
4
5  test();
```

5. Para añadir este archivo al repositorio local haremos uso de 2 comandos, *git add .* y *git commit -m ""* el primer comando simplemente añadirá a HEAD nuestros cambios, y el segundo comando nos permitirá añadir un comentario a esos cambios, por ejemplo "First commit".

```
iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
● $ git add .

iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
● $ git commit -m "First commit"
[master (root-commit) 7d1d193] First commit
 1 file changed, 5 insertions(+)
 create mode 100644 src/main.ts

iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
○ $
```

6. En este punto ya estamos listos para añadir los cambios del repositorio local al repositorio remoto, pero nos falta el último paso de la vinculación, para completar todo este proceso usaremos el comando *git remote add origin "url"* (donde "url" será la url de nuestro repositorio remoto) el cual nos permitirá establecer una vinculación para que cuando nosotros decidamos añadir el contenido al repositorio remota use *origin* como destino final.

```
iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
● $ git remote add origin https://github.com/S4nchzz/test-repositorio

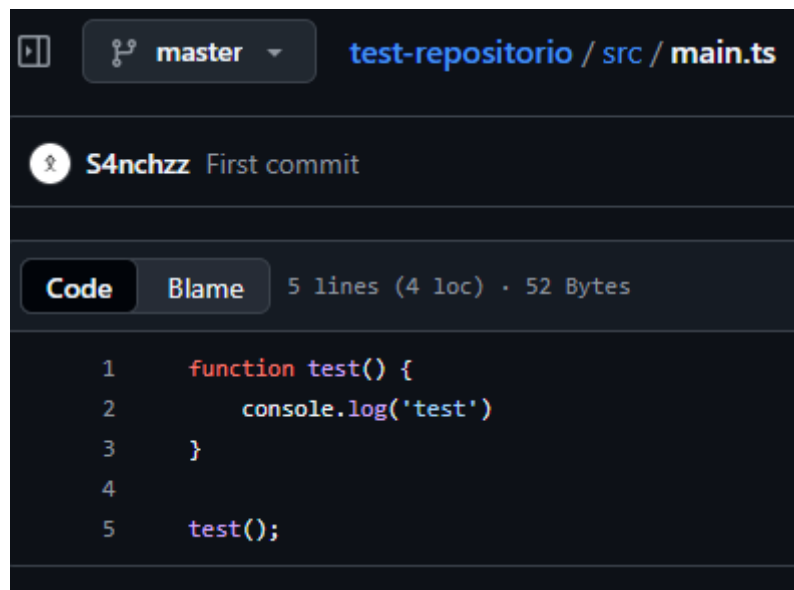
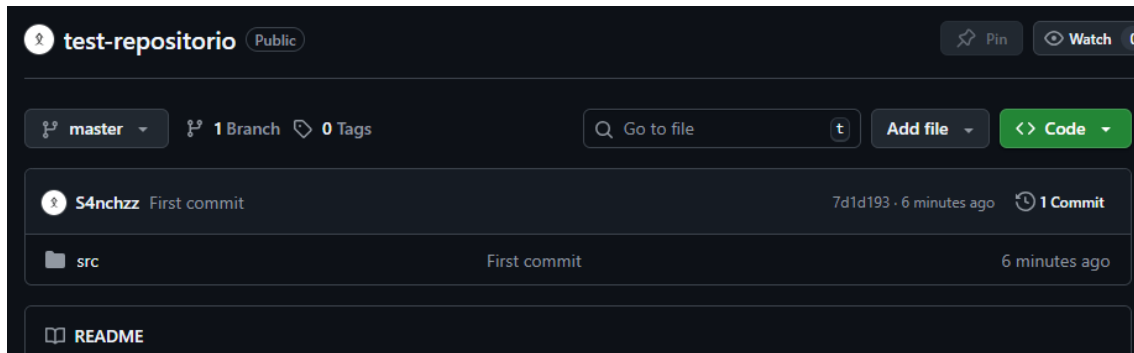
iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
○ $
```

7. Por ultimo, subiremos los cambios con *git push origin master*

```
iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
● $ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 297 bytes | 297.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/S4nchzz/test-repositorio
 * [new branch]      master -> master

iyansd@A210P11 MINGW64 /c/Users/iyansd/Desktop/test (master)
○ $
```

8. Si ahora vamos al repositorio remoto en github podremos ver que ahora nos aparece la carpeta src que creamos en local y dentro esta el archivo main.ts



*FIN DEL DOCUMENTO*