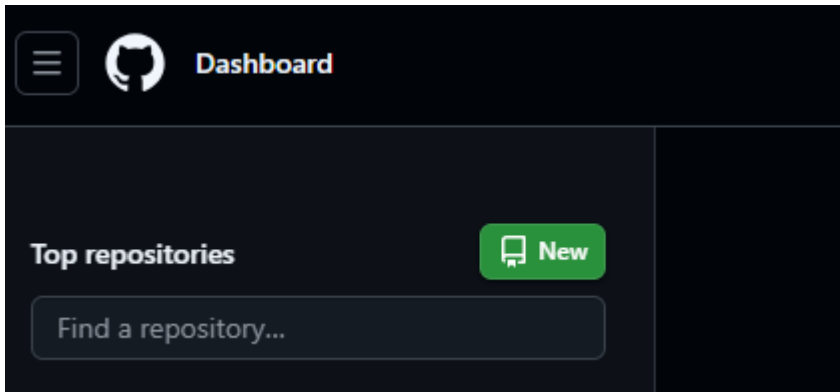


COMO UTILIZAR GIT Y GITHUB




Creamos el repositorio: aquí se crea con el nombre de la carpeta que luego estará en nuestra máquina.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 alforhanz ▾



Repository name *

Resportes

✔ Resportes is available.

Great repository names are short and memorable. Need inspiration? How about **fluffy-system** ?

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

- ☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: **None** ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

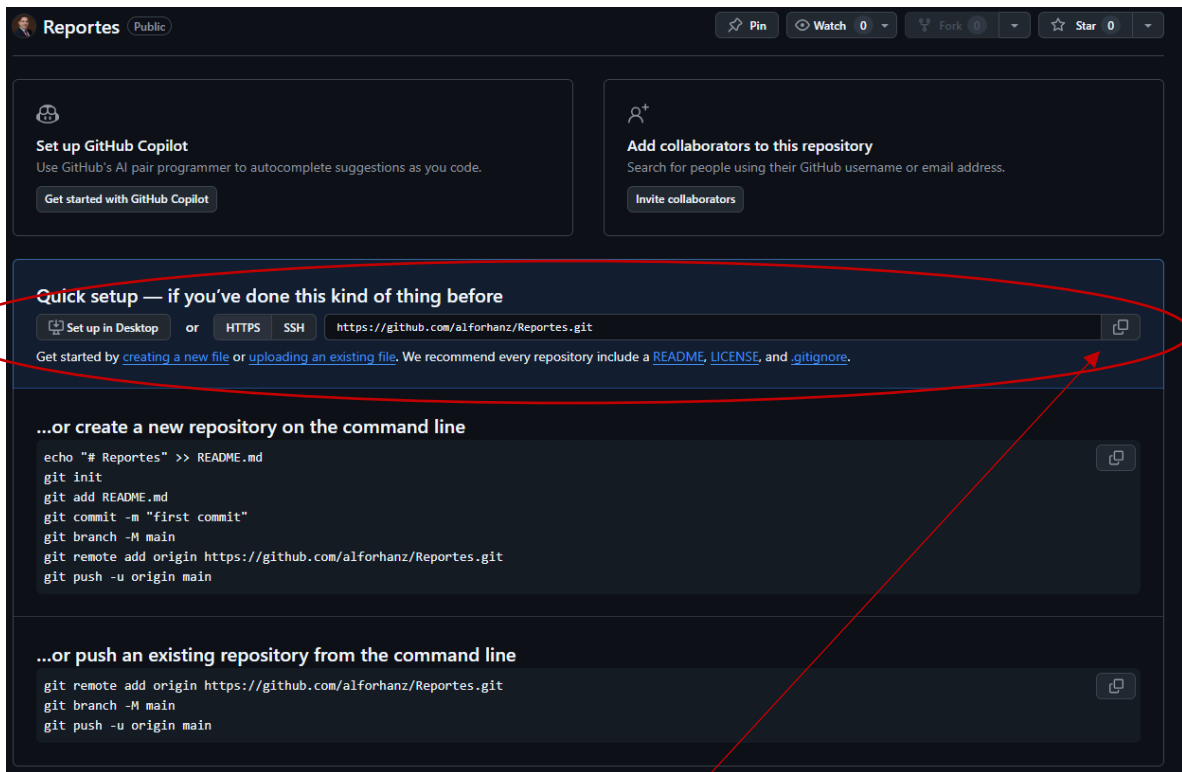
License: **None** ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository

Damos clic en el botón créate repository.

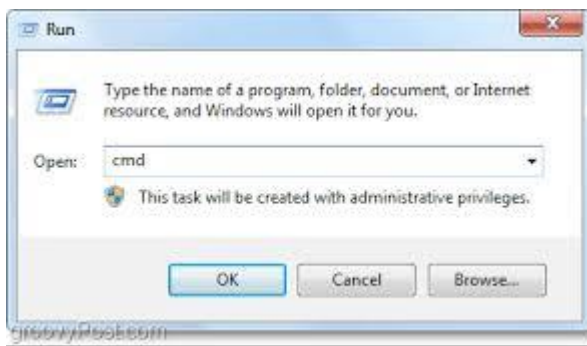


Copiamos el enlace dando clic en el icono de copiar.

Abrimos la consola, Windows + R:



en la ventana emergente escribimos cmd



Cuando estemos en la consola utilizando el comando dir vamos a poder ver en que directorio o carpeta estamos ubicados y a que carpeta vamos a ir.

```
C:\> C:\WINDOWS\system32\cmd.exe
D:\CLSA-512>dir
El volumen de la unidad D es Datos(SSD)
El número de serie del volumen es: FE58-8DA1

Directorio de D:\CLSA-512

03/02/2024  11:46 a. m.      <DIR>          .
03/02/2024  11:46 a. m.      <DIR>          ..
04/25/2025  04:37 p. m.      <DIR>          Descargas
05/10/2025  11:13 a. m.      <DIR>          Documentos
05/05/2025  09:52 a. m.      <DIR>          Escritorio
02/05/2025  03:02 p. m.      <DIR>          Imagenes
03/02/2024  11:49 a. m.      <DIR>          Musica
03/08/2024  12:47 a. m.      <DIR>          Videos
           0 archivos             0 bytes
           8 dirs  29,296,713,728 bytes libres

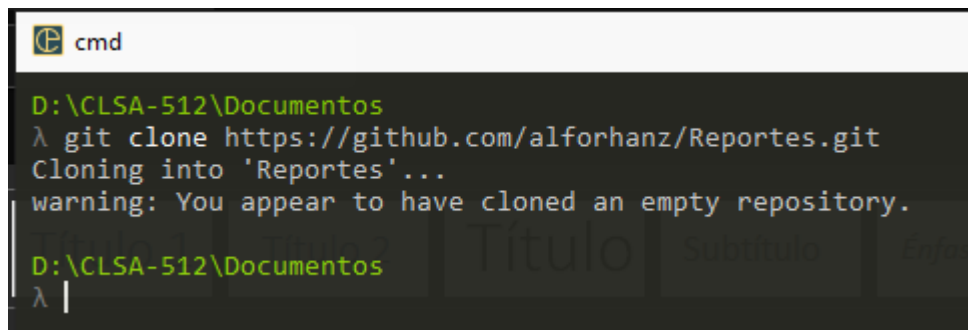
D:\CLSA-512>
```

En nuestro caso podemos utilizar el directorio Documentos, así que vamos a usar el comando cd o change directory, para cambiar de directorio o carpeta.

```
D:\CLSA-512>cd Documentos  
D:\CLSA-512\Documentos>_
```

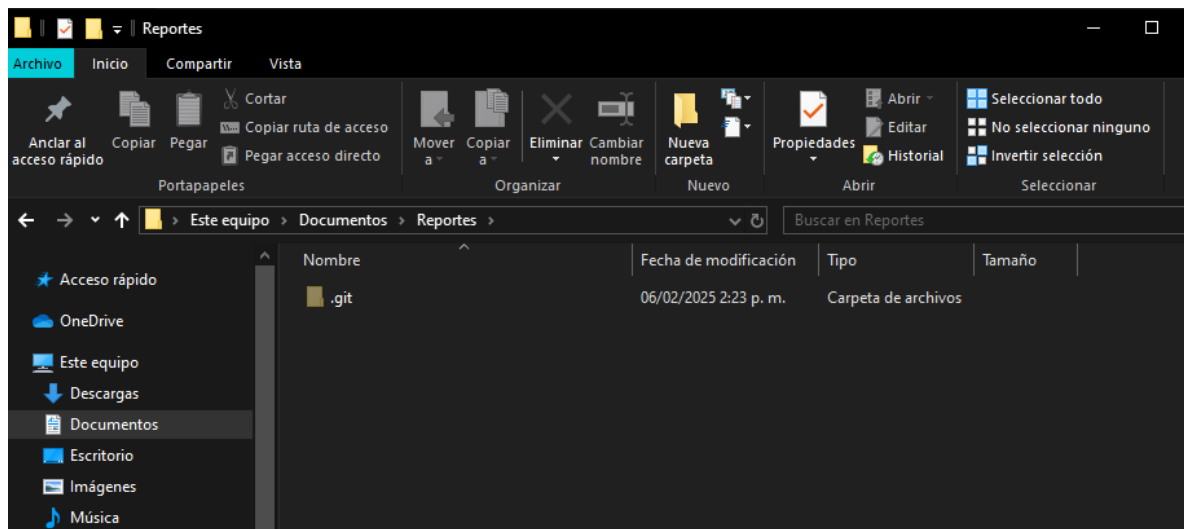
Ahora usamos el comando git clone y pegamos el enlace de nuestro repositorio remoto.

```
D:\CLSA-512\Documentos>git clone https://github.com/alforhantz/Reportes.git
```

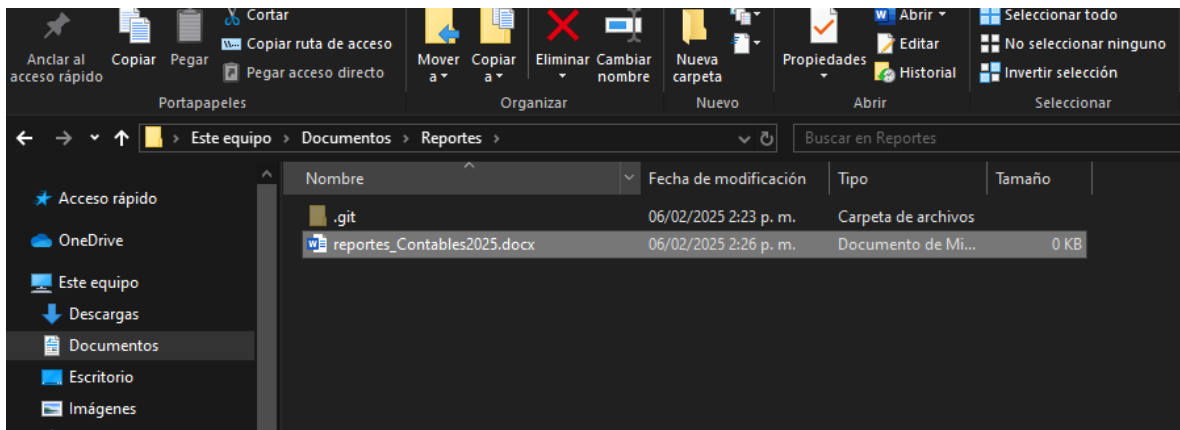


```
cmd  
D:\CLSA-512\Documentos  
λ git clone https://github.com/alforhantz/Reportes.git  
Cloning into 'Reportes'...  
warning: You appear to have cloned an empty repository.  
D:\CLSA-512\Documentos  
λ |
```

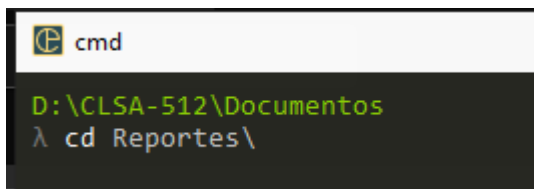
Se crea la carpeta del repositorio.



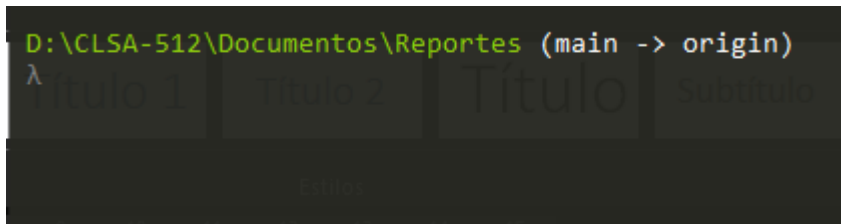
Aquí guardaremos los diferentes archivos de los que necesitemos mantener un seguimiento.



Cuando terminamos de trabajar en el documento hayamos guardado los cambios realizados, vamos nuevamente a la consola cmd y entramos a nuestro repositorio utilizando el comando cd.



Veremos algo como esto:



Significa que estamos dentro del directorio o carpeta del repositorio que tenemos en nuestra máquina (repositorio local).

Utilizando el comando **git status**: aquí vamos a ver los archivos que se han agregado o modificado.

```

D:\CLSA-512\Documentos\Reportes (main -> origin)
λ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    reportes_Contables2025.docx

nothing added to commit but untracked files present (use "git add" to track)
D:\CLSA-512\Documentos\Reportes (main -> origin)
λ

```

el sistema nos indica que podemos utilizar la opción git add . para agregar los archivos con los cambios realizados a nuestro stage-Area.

Una vez ejecutamos el comando git add . verificamos con git status si se agregó este al stage-área.

```

D:\CLSA-512\Documentos\Reportes (main -> origin)
λ git add .

D:\CLSA-512\Documentos\Reportes (main -> origin)
λ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   reportes_Contables2025.docx

```

Luego vamos a realizar el commit y agregar mis cambios y archivos a mi repositorio local. usando el comando git commit -m "comentario...", reemplazamos donde dice comentario por alguna observación del cambio realizado, esto es preferible que sea de manera corta.

```

cmd
D:\CLSA-512\Documentos\Reportes (main -> origin)
λ git commit -m "Subiendo los archivos a mi repositorio"
[main (root-commit) a07a050] Subiendo los archivos a mi repositorio
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 reportes_Contables2025.docx

```

Si hacemos git status para verificar el estado de nuestro repositorio, nos deberá mostrar lo siguiente:

```
D:\CLSA-512\Documentos\Reportes (main -> origin)
λ git status
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
```

Ahora hacemos **git push** para subir los cambios de nuestro repositorio local en nuestra máquina, a nuestro repositorio remoto en el servidor de github.

```
D:\CLSA-512\Documentos\Reportes (main -> origin)
λ git push
"C:/Users/CLSA-51/cmdr/vendor/git-for-windows/mingw64/bin/git-credential-manager.exe" get: line 1: C:/Users/CLSA-51/cmdr/vendor/git-for-windows/mingw64/bin/git-credential-manager.exe: No such file or directory
warning: could not find UI helper 'GitHub.UI'
Select an authentication method for 'https://github.com/alforhanz/Reportes.git':
  1. Web browser (default)
  2. Device code
  3. Personal access token
option (enter for default):
```

```
D:\CLSA-512\Documentos\Reportes (main -> origin)
λ git push
"C:/Users/CLSA-51/cmdr/vendor/git-for-windows/mingw64/bin/git-credential-manager.exe" get: line 1: C:/Users/CLSA-51/cmdr/vendor/git-for-windows/mingw64/bin/git-credential-manager.exe: No such file or directory
warning: could not find UI helper 'GitHub.UI'
Select an authentication method for 'https://github.com/alforhanz/Reportes.git':
  1. Web browser (default)
  2. Device code
  3. Personal access token
option (enter for default): 1
info: please complete authentication in your browser...
"C:/Users/CLSA-51/cmdr/vendor/git-for-windows/mingw64/bin/git-credential-manager.exe" store: line 1: C:/Users/CLSA-51/cmdr/vendor/git-for-windows/mingw64/bin/git-credential-manager.exe: No such file or directory
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), 256 bytes | 128.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/alforhanz/Reportes.git
 * [new branch]      main -> main
```

Repitiendo este proceso, podemos hacer un seguimiento a los cambios en un documento o varios archivos.

RESUMEN

git status: verificamos el status de nuestro repositorio, si existen cambios pendientes.

git add . (git add punto): agregamos los archivos o cambios al repositorio remoto.

git commit -m "comentario": comentamos y realizamos el commit para guardar los cambios en nuestro repositorio local.

git push: subimos los cambios a nuestro repositorio remoto.

Notas:

si en algún punto tienes problemas y necesitas regresar a una versión del documento o archivo, en la que todo este funcionando, porque el documento tiene errores o porque se guardaron cambios con errores, podemos utilizar los siguientes comandos:
git log: esto devuelve todos los commit del repositorio.

cada vez que se realiza un cambio, se genera una cadena única, para identificar el cambio que hemos realizado en el repositorio, archivo o documento.

```
D:\CLSA-512\Documentos\Reportes (main -> origin)
λ git log
commit 7127e66da8b2e28943de69964a198e03ba523854 (HEAD -> main, origin/main)
Author: alforhanz <alforhanz@users.noreply.github.com>
Date: Mon Jun 2 16:28:20 2025 -0500
    cambios del documento.

commit a07a050c6ad078e6629ac9c9f51d90fa1c8c9c9c
Author: alforhanz <alforhanz@users.noreply.github.com>
Date: Mon Jun 2 15:10:05 2025 -0500
    Subiendo los archivos a mi repositorio
```

copiamos la cadena a07a050c6ad078e6629ac9c9f51d90fa1c8c9c9c
y utilizamos el comando **git reset --hard a07a050c6ad078e6629ac9c9f51d90fa1c8c9c9c**

¡Advertencia importante!

Este comando:

- **Elimina todos los cambios posteriores al commit especificado** en tu directorio de trabajo y en el historial.
- Si ya hiciste push de los commits posteriores al remoto (por ejemplo en GitHub), tendrás que forzar el push para reflejar este cambio:

git push origin main --force.

