

Manual de inicios en Git y GitHub

1. Git	Pg. 2
2. GitHub	Pg. 3
3. Prueba Pull	Pg. 8
4. Prueba Clone	Pg. 10
5. Prueba Gitignore	Pg. 11

1. Git

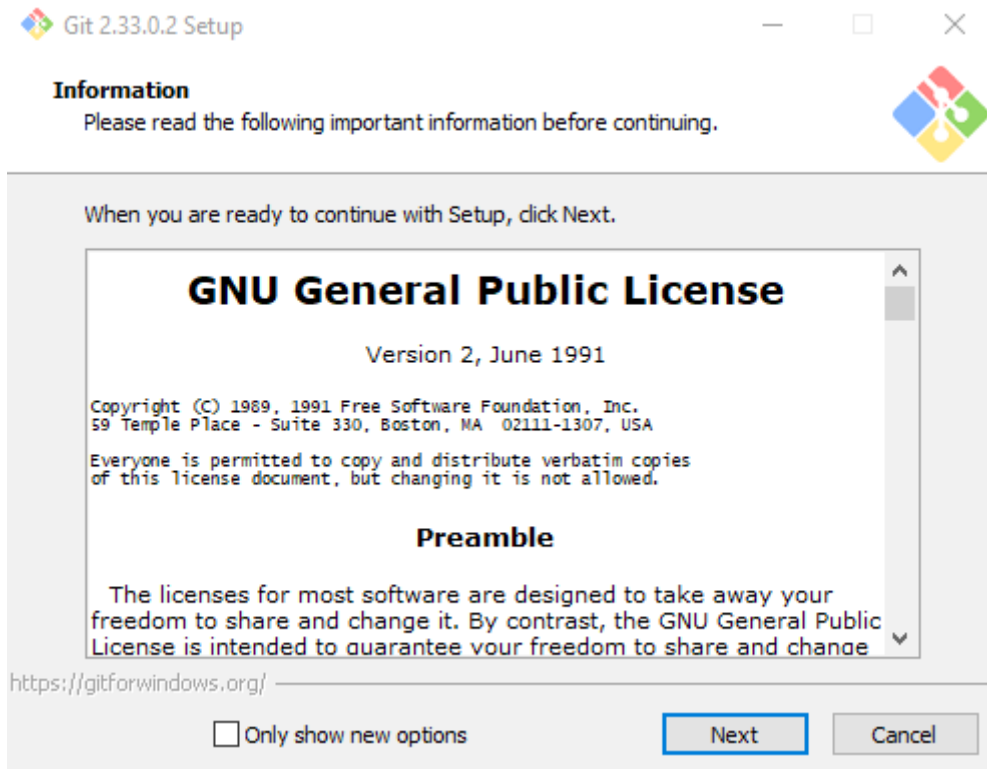
Para comenzar debemos ir a la página oficial (<https://git-scm.com/>) y pulsar sobre la pantallita azul que dice **Download for Windows**.



Ahora nos habrá redirigido a la página de descargas y debería de comenzar automáticamente la descarga del instalador.



Cuando la descarga finalice, abriremos el ejecutable con doble click izquierdo y pasaremos sobre las diferentes etapas sin cambiar ninguna opción pulsando en **Next**.

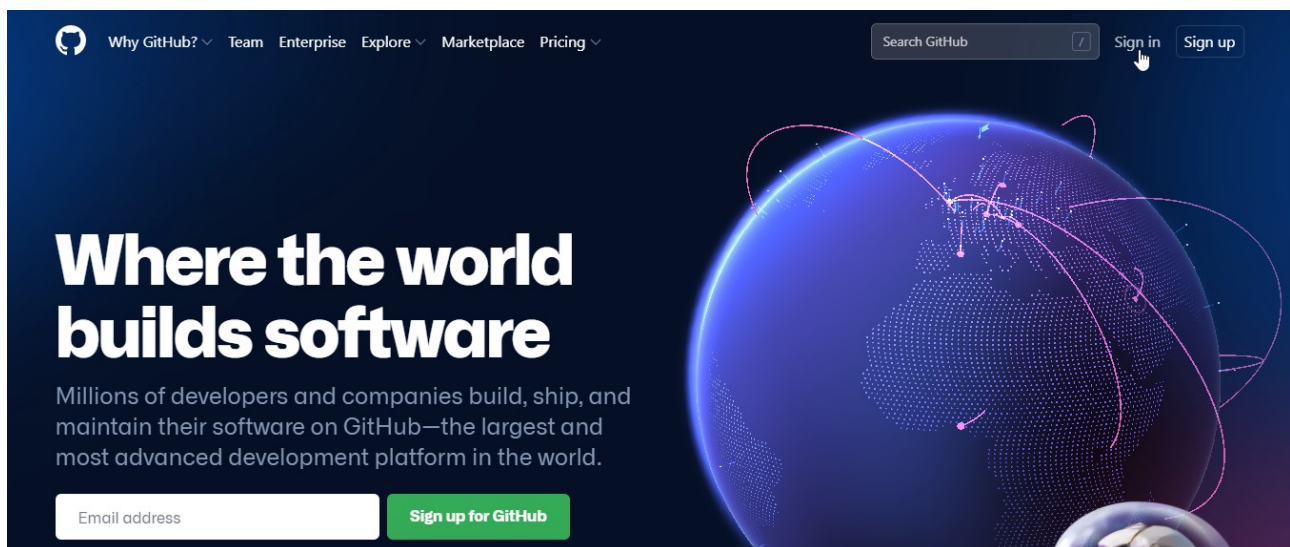


Tras acabar con la instalación de Git, es hora de comenzar a utilizarlo.

2. GitHub

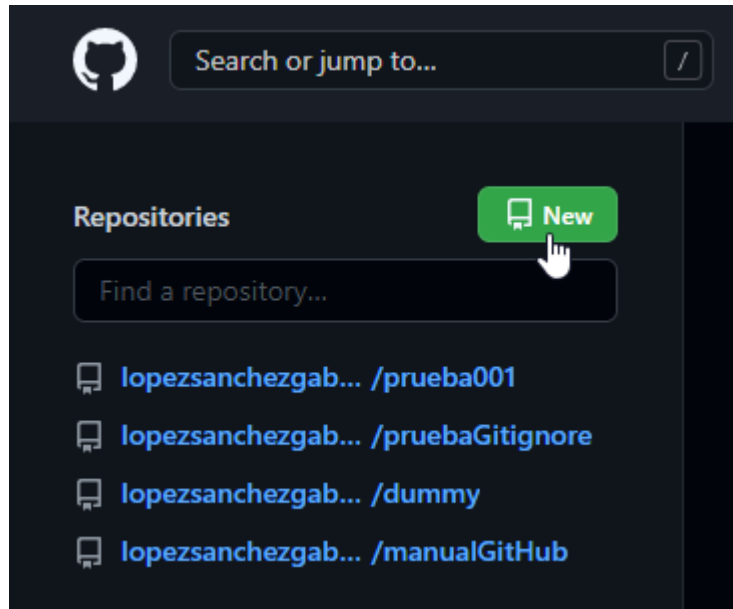
Con Git instalado procederemos a una sencilla introducción a GitHub, nuestra plataforma de repositorios en la nube.

Lo primero será crear una cuenta o iniciar sesión por medio de Google.



Al iniciar sesión podremos ver a la izquierda un panel con los repositorios, si la cuenta es nueva simplemente no aparecerá ninguno.

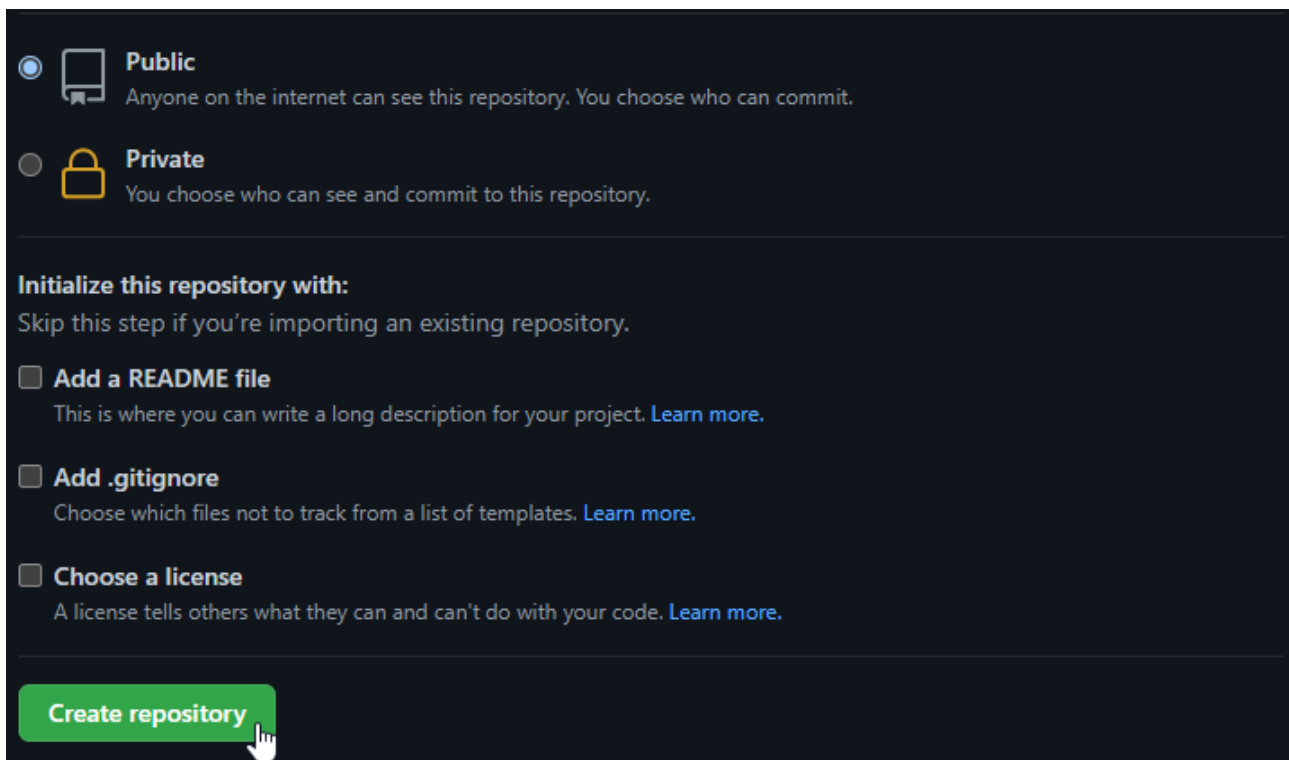
Para comenzar, se deberá crear un repositorio al que subir nuestro trabajo.



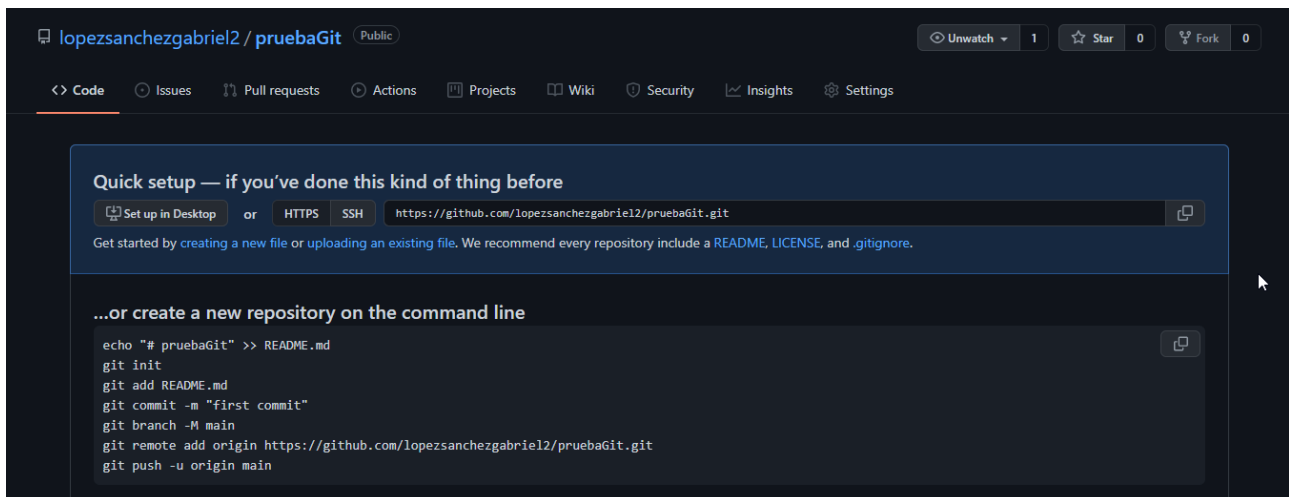
Se le dará el nombre que deseemos, siempre y cuando no sea idéntico a otro que hayamos creado, ya que se guardan bajo el nombre de nuestro perfil, no interferirá con el repositorio de otro usuario.

A screenshot of the 'Create a new repository' form on GitHub. The title 'Create a new repository' is at the top. Below it is a subtitle: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' The form has two main sections. The first section is for the repository details, with 'Owner' and 'Repository name' as labels. The 'Owner' dropdown shows 'lopezsanchezgabriel2'. The 'Repository name' text input contains 'pruebaGit' and has a green checkmark icon to its right. Below this, a message says 'Great repository names are short and <pruebaGit is available. inspiration? How about ubiquitous-couscous?'. The second section is for the repository's visibility, with 'Description (optional)' as a label. There are two radio buttons: 'Public' (selected) and 'Private'. The 'Public' option has a description: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option has a description: 'You choose who can see and commit to this repository.'

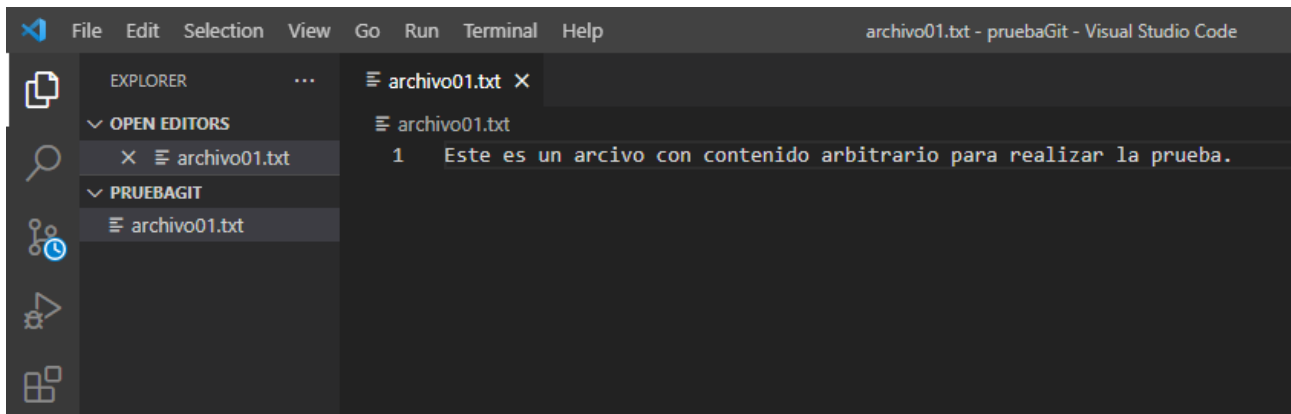
Como sólo estamos viendo su funcionamiento procederemos a crearlo.



Al visualizar el repositorio, podemos comprobar que está vacío y nos proporciona varios métodos para ponerlo en uso. Nosotros utilizaremos algunos de los comandos de la ventana gris.

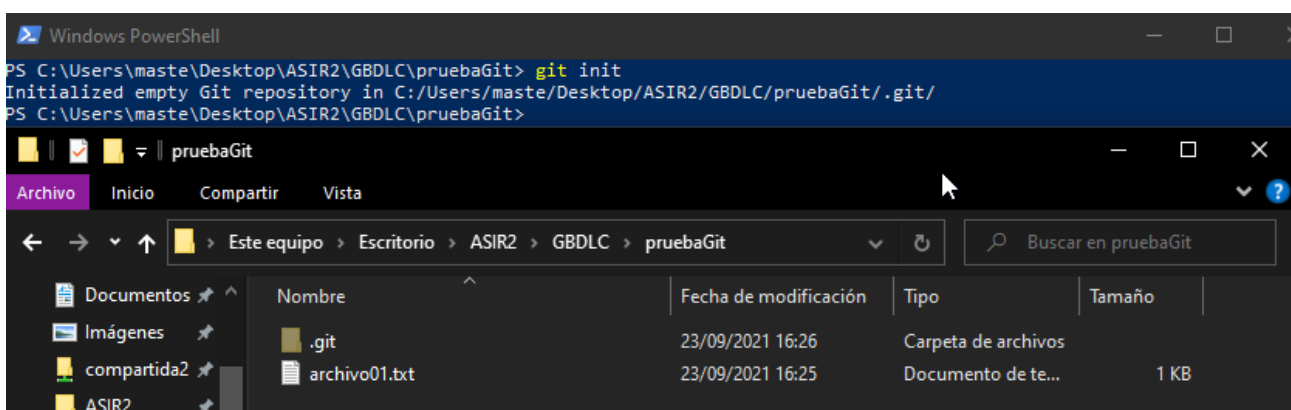
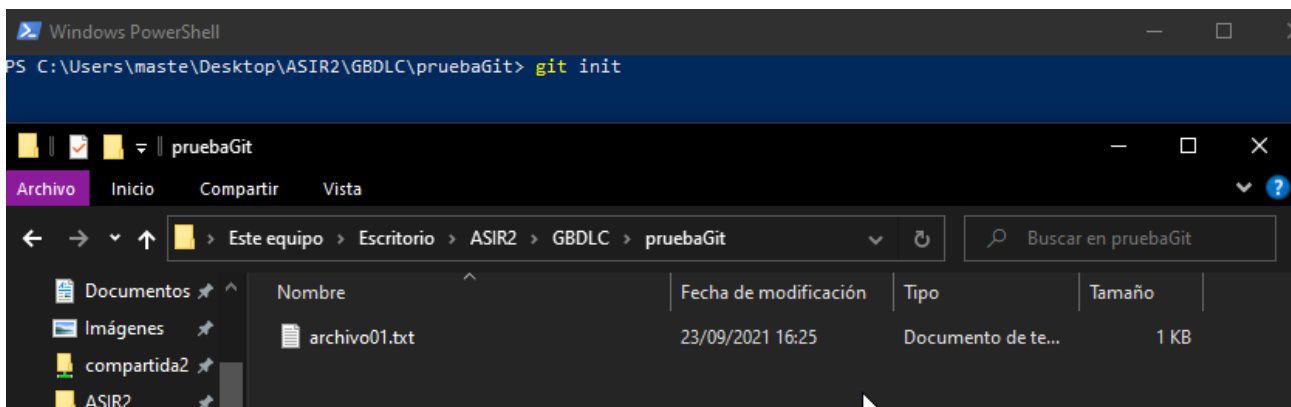


Lo siguiente es crear un archivo, el cual subiremos al repositorio.



A continuación, abriremos una consola de PowerShell y cambiaremos su directorio al de nuestra carpeta con el archivo anterior.

Como se puede comprobar, sólo existe un archivo antes de lanzar la instrucción **git init**, que creará una carpeta oculta llamada **.git** para trabajar el programa y que no deberemos tocar.



Ahora que lo hemos iniciado usaremos **git add** . para añadir todos los archivos y carpetas en el directorio actual (sólo un archivo en este caso).

Si introducimos **git status** nos mostrará el estado de nuestros archivos.

Con **git commit** realizaremos una primera validación de los archivos a subir y entrará en la “etapa” de montaje.

Le indicaremos que se subirán en la rama principal de repositorio con ***git branch -M main***. Configuraremos su destino con ***git remote add origin <URL del repositorio>***. Por último la orden ***Push*** que volcará en la nube nuestro repositorio.

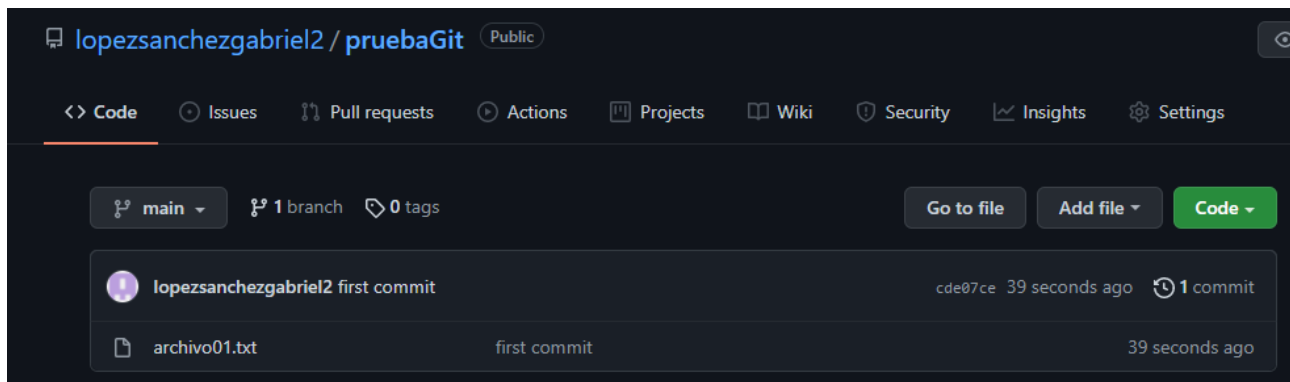
```
Windows PowerShell
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGit> git init
Initialized empty Git repository in C:/Users/maste/Desktop/ASIR2/GBDLC/pruebaGit/.git/
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGit> git add .
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGit> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   archivo01.txt

PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGit> git commit -m "first commit"
[master (root-commit) cde07ce] first commit
 1 file changed, 1 insertion(+)
   create mode 100644 archivo01.txt
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGit> git branch -M main
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGit> git remote add origin https://github.com/lopezsanchezgabriel2/pruebaGit.git
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGit> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 285 bytes | 285.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/lopezsanchezgabriel2/pruebaGit.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGit>
```

Si actualizamos nuestra página web del repositorio podremos comprobar que ha funcionado.



3. Prueba Pull

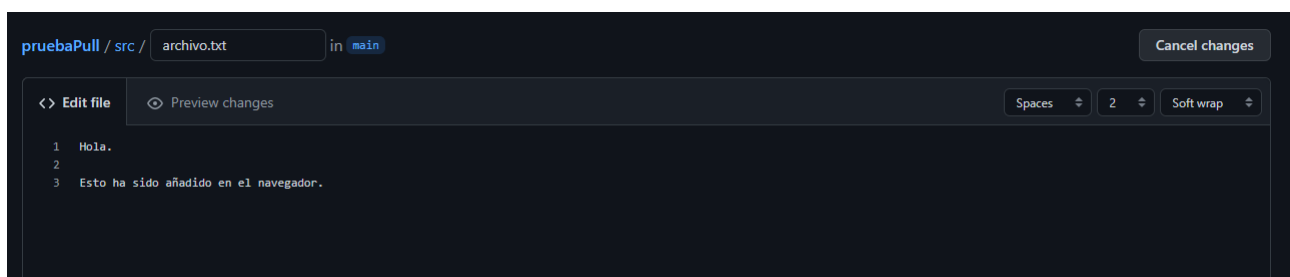
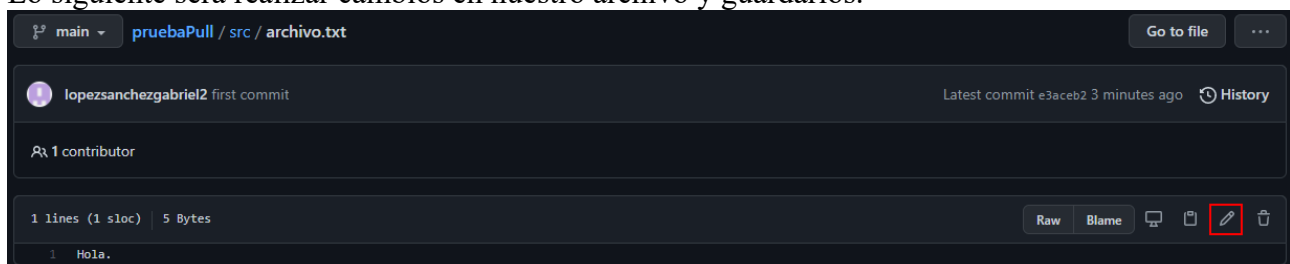
Ahora veremos cómo actualizar nuestros archivos locales si se han modificado en nuestro repositorio en la nube.

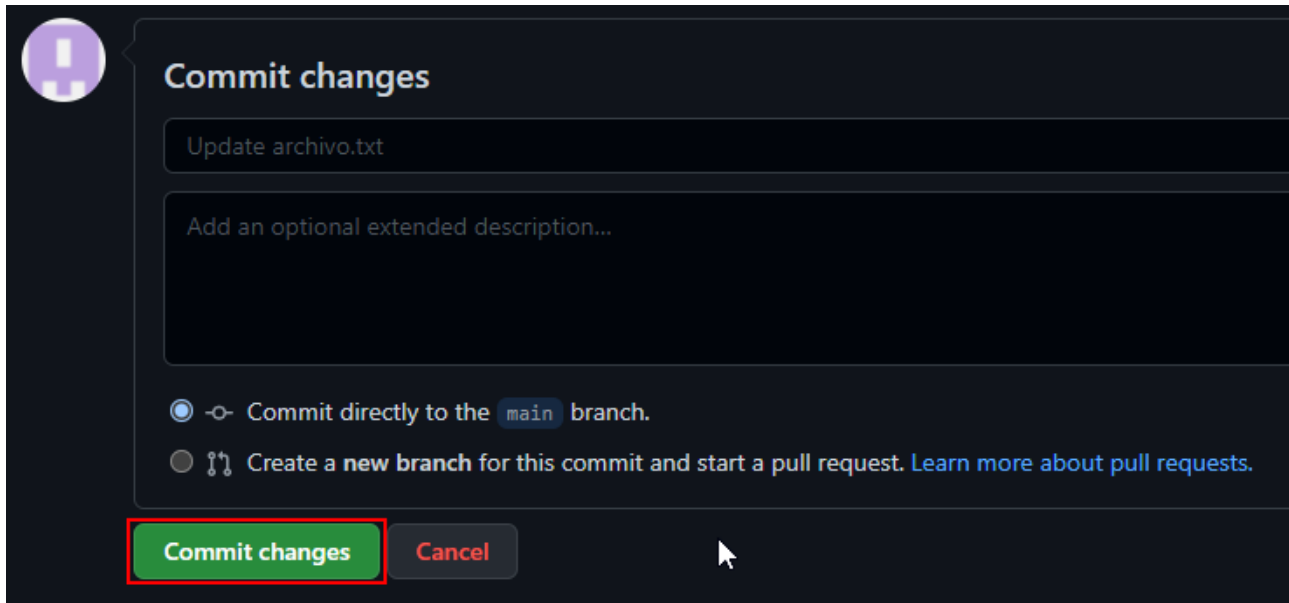
Comenzaremos realizando un repositorio y subiendo un archivo como en el apartado anterior.

```
src > ≡ archivo.txt
1   Hola.
```

```
Windows PowerShell
PS C:\Users\maste> cd C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git init
Initialized empty Git repository in C:/Users/maste/Desktop/ASIR2/GBDLC/pruebaPull/.git/
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git add .
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git commit -m "first commit"
[master (root-commit) e3aceb2] first commit
2 files changed, 6 insertions(+)
 create mode 100644 README.md
 create mode 100644 src/archivo.txt
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git branch -M main
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git remote add origin https://github.com/lopezsanchezgabriel2/pruebaPull.git
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 462 bytes | 462.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/lopezsanchezgabriel2/pruebaPull.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull>
```

Lo siguiente será realizar cambios en nuestro archivo y guardarlos.





Si tratásemos de subir algún cambio al repositorio, nos dará un error y una advertencia de que hay información que no corresponde con la de nuestro equipo y que deberíamos integrarla antes de subir la nuestra.

```
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git push -u origin main
To https://github.com/lopezsanchezgabriel2/pruebaPull.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/lopezsanchezgabriel2/pruebaPull.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.
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull>
```

Así que ejecutaremos **git pull** para actualizar nuestro trabajo en el equipo.

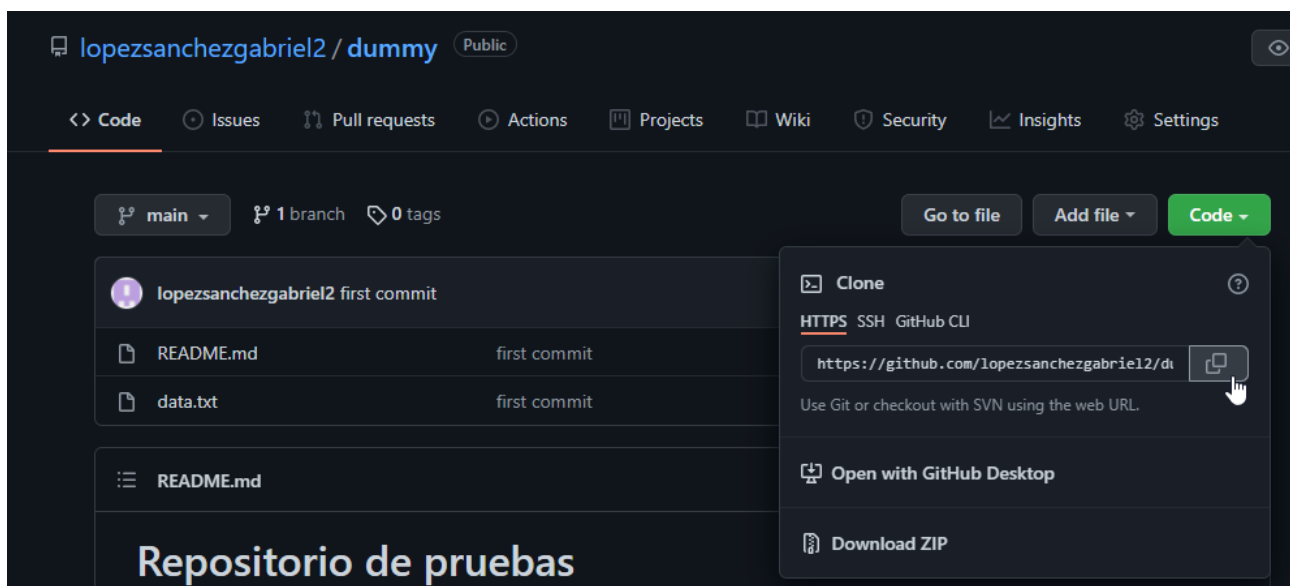
```
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull> git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 752 bytes | 27.00 KiB/s, done.
From https://github.com/lopezsanchezgabriel2/pruebaPull
   e3aceb2..8791eda  main      -> origin/main
Updating e3aceb2..8791eda
Fast-forward
 src/archivo.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaPull>
```

Si vamos al archivo, veremos que los cambios realizados previamente en la web también aparecen en nuestro archivo local.

```
src > ≡ archivo.txt
1  Hola.
2
3  Esto ha sido añadido en el navegador.
4
```

4. Prueba Clone

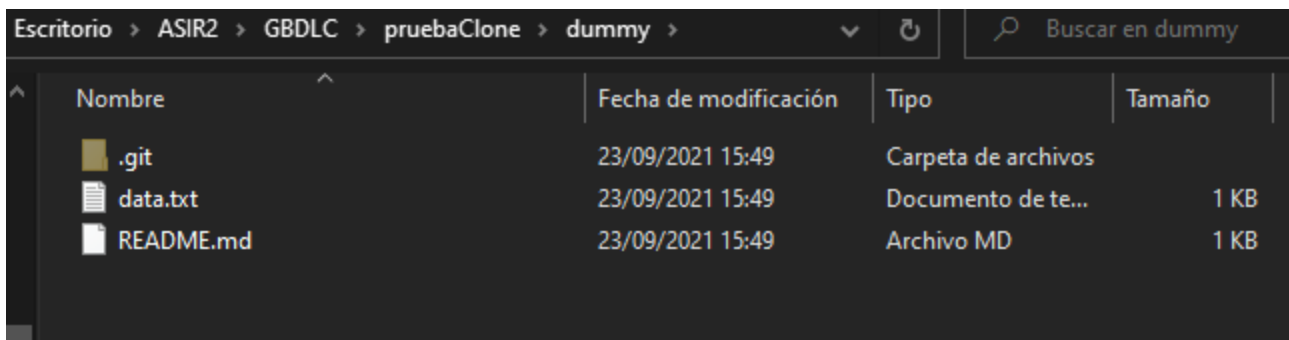
Para ésta prueba copiaremos la URL que aparece en la ventana al pulsar en el botón verde **Code**.



Ejecutaremos el comando *git clone* <URL copiada> en el directorio donde deseamos copiar el repositorio.

```
Windows PowerShell
PS C:\Users\maste\Desktop\ASIR2\GBDLC> cd C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaClone
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaClone> git clone https://github.com/lopezsanchezgabriel2/dummy.git
Cloning into 'dummy'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaClone>
```

Como podemos observar, copia el repositorio al completo, no los archivos de su interior.

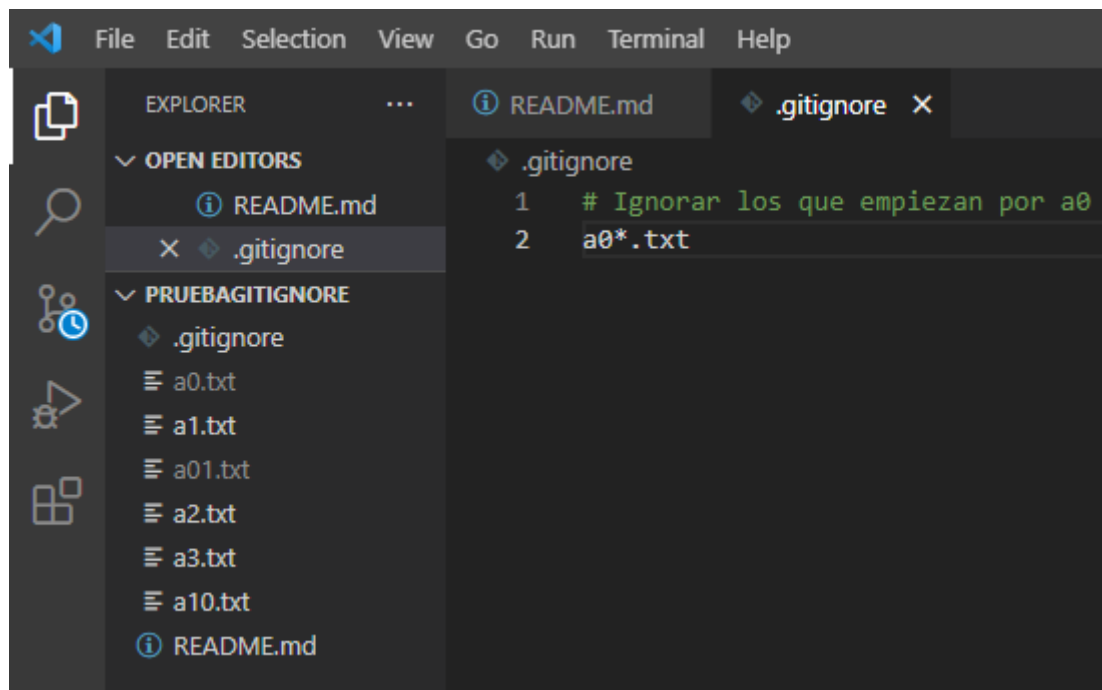


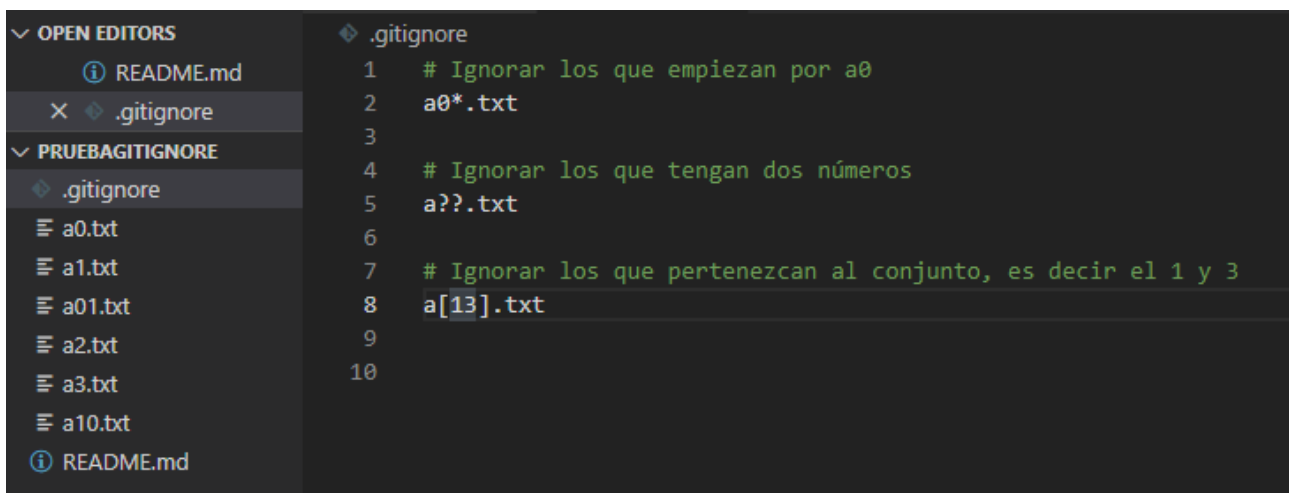
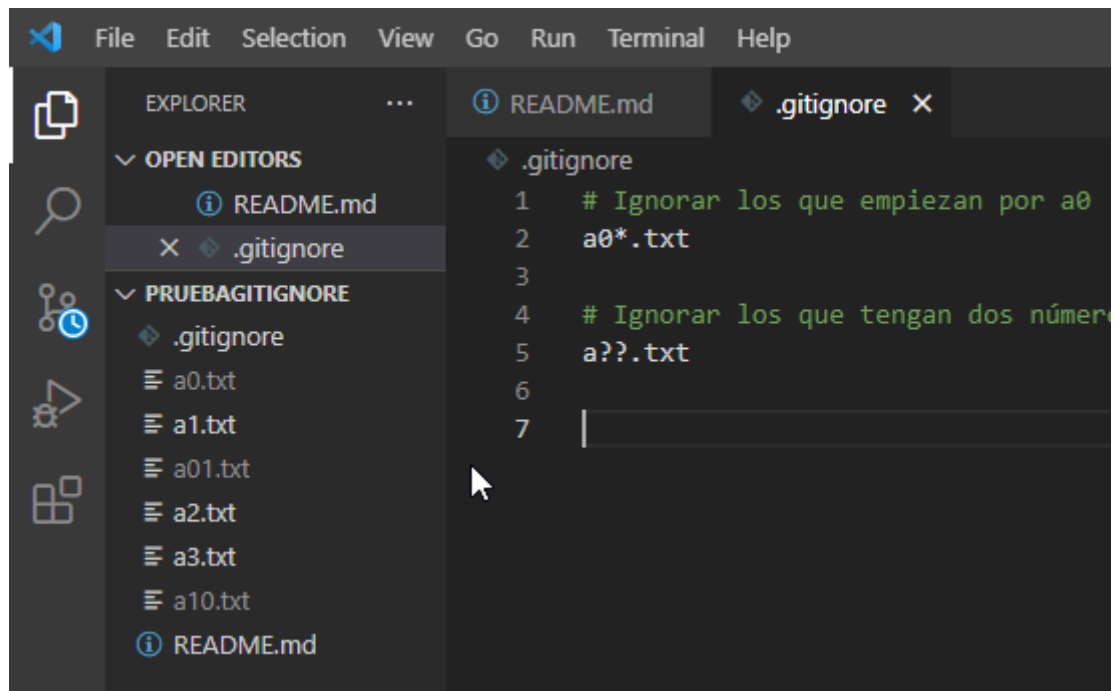
Nombre	Fecha de modificación	Tipo	Tamaño
.git	23/09/2021 15:49	Carpeta de archivos	
data.txt	23/09/2021 15:49	Documento de te...	1 KB
README.md	23/09/2021 15:49	Archivo MD	1 KB

5. Prueba Gitignore

Necesitaremos varios archivos de entre los cuales, haremos que sólo **a2.txt** sea volcado a la nube.

Para ello usaremos un archivo llamado **.gitignore** en el que podremos utilizar comodines (?-un carácter o *-uno o varios caracteres) en forma de filtro.





```
Windows PowerShell
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGitignore> git init
Initialized empty Git repository in C:/Users/maste/Desktop/ASIR2/GBDLC/pruebaGitignore/.git/
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGitignore> git add .
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGitignore> git commit -m "first commit"
[master (root-commit) 7fd62b1] first commit
 3 files changed, 9 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 README.md
 create mode 100644 a2.txt
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGitignore> git branch -M main
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGitignore> git remote add origin https://github.com/lopezsanchezgabriell2/pru
ebaGitignore.git
PS C:\Users\maste\Desktop\ASIR2\GBDLC\pruebaGitignore> git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 804 bytes | 804.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/lopezsanchezgabriell2/pruebaGitignore.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

lopezsanchezgabriell2 / pruebaGitignore (Public)

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

lopezsanchezgabriell2 first commit 7fd62b1 1 minute ago 1 commit

.gitignore	first commit	1 minute ago
README.md	first commit	1 minute ago
a2.txt	first commit	1 minute ago

