

Instalación y explicación de git y github.

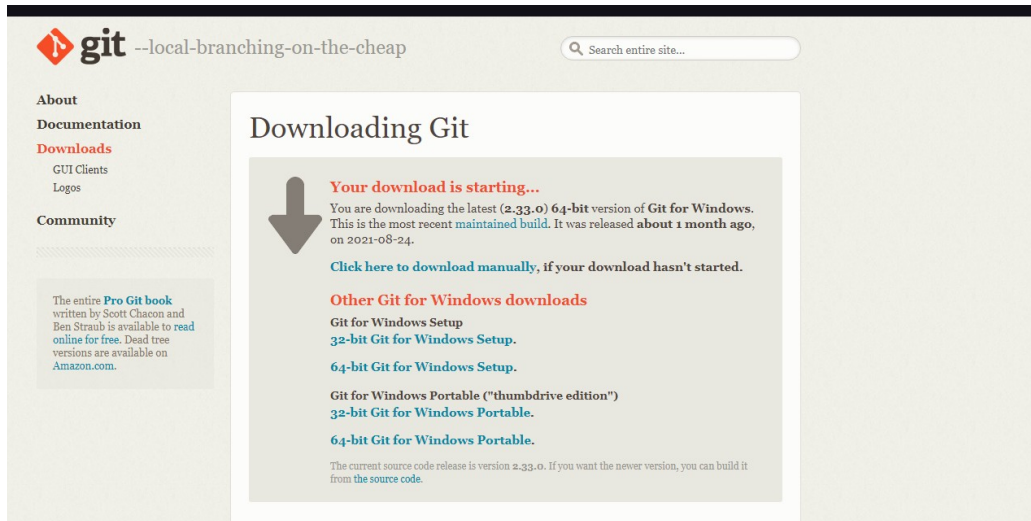
-Gestión de Bases de Datos no SQL-

índice

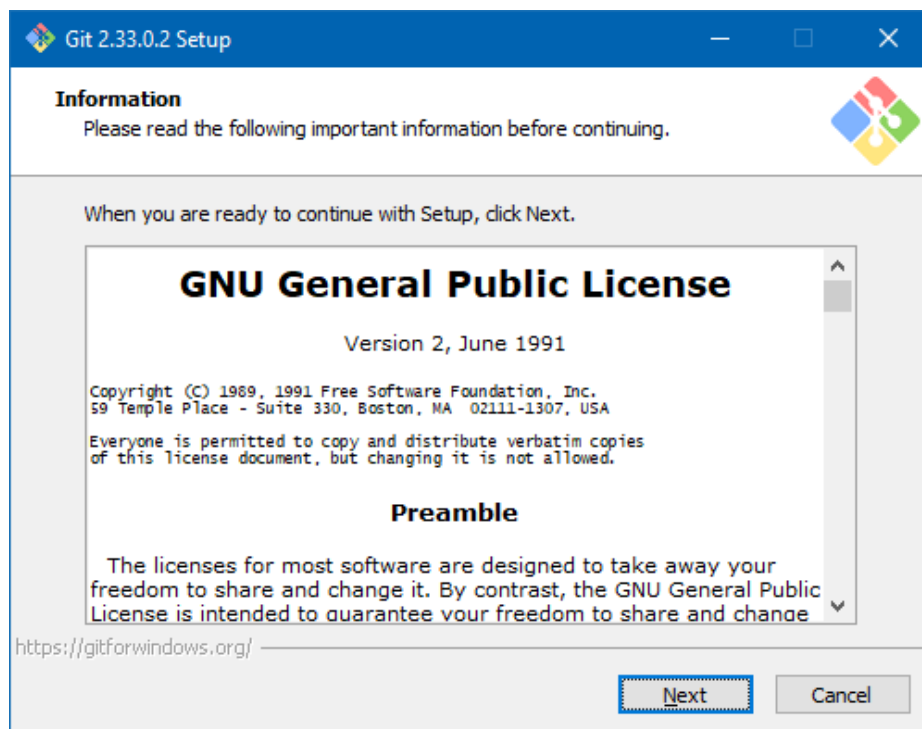
1.Instalación de git.....	2
2.Crear usuario y repositorio en Github.....	6
3.Como crear un repositorio local y subirlo a Github.....	8
4.Practicando con algunos comandos.....	12
4.1.Modificación de archivos.....	12
4.2.Clonación del repositorio.....	14
4.3.Actualización local de un repositorio.....	15
4.4.Creación de readme.....	16

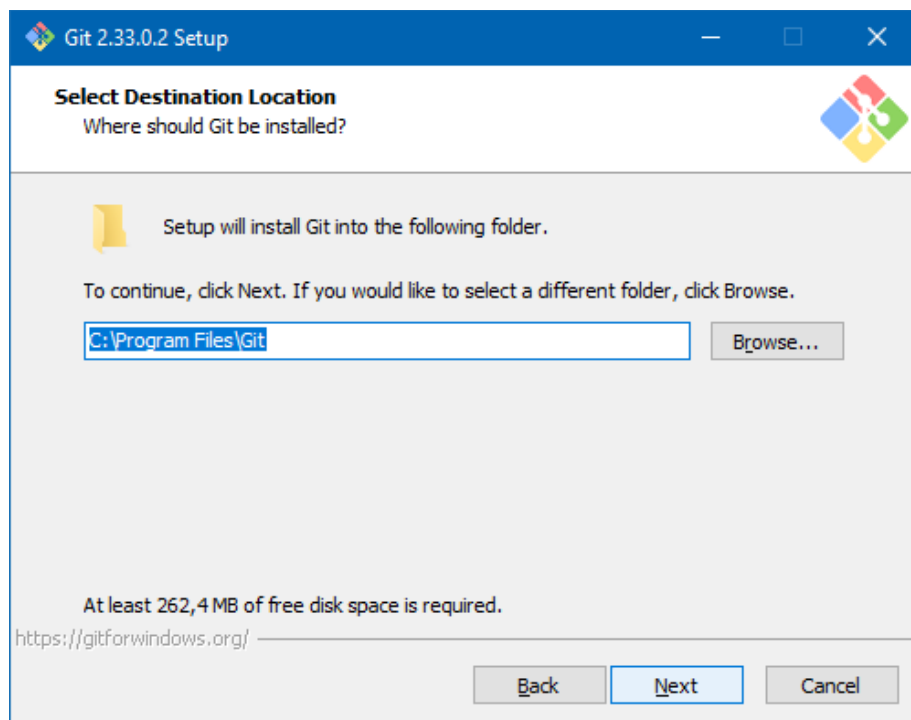
1.Instalación de git.

Para instalar git nos dirigiremos a su página oficial <https://git-scm.com/download/win> donde podremos descargarlo.

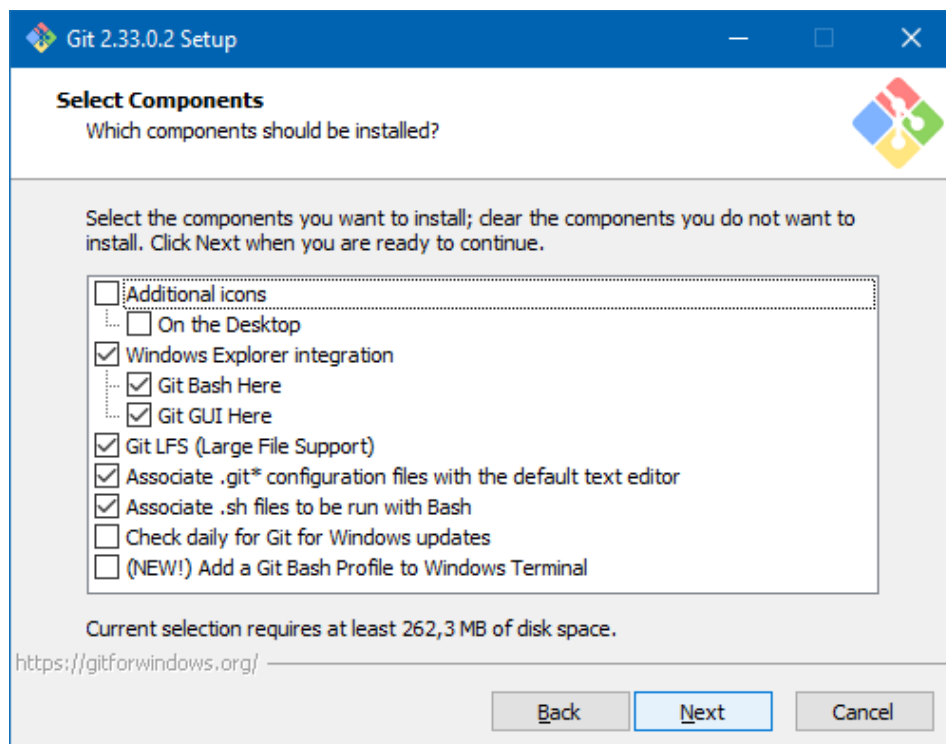


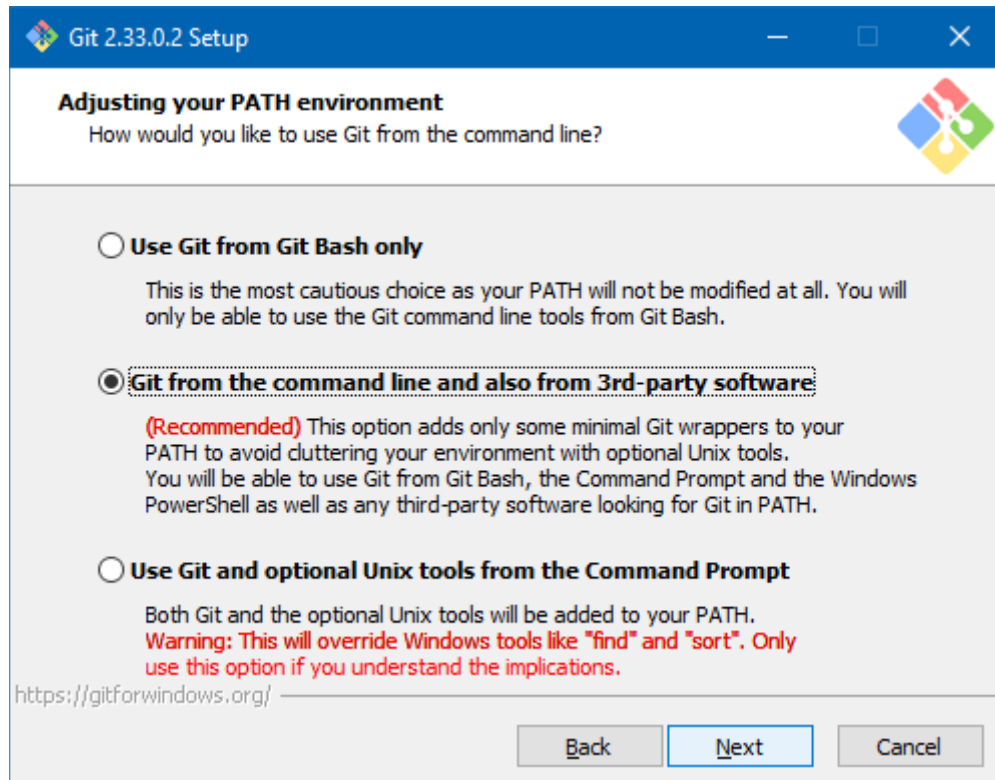
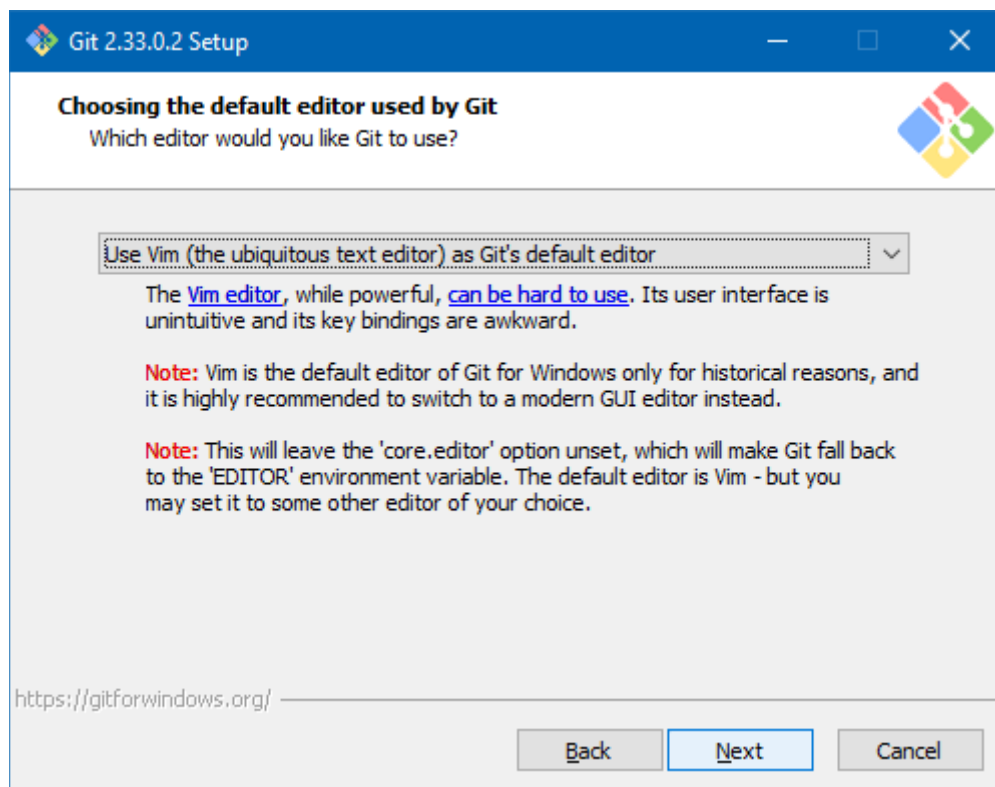
Una vez descargado, debemos ejecutar el instalador, pulsar en next y elegir la ruta donde instalaremos.

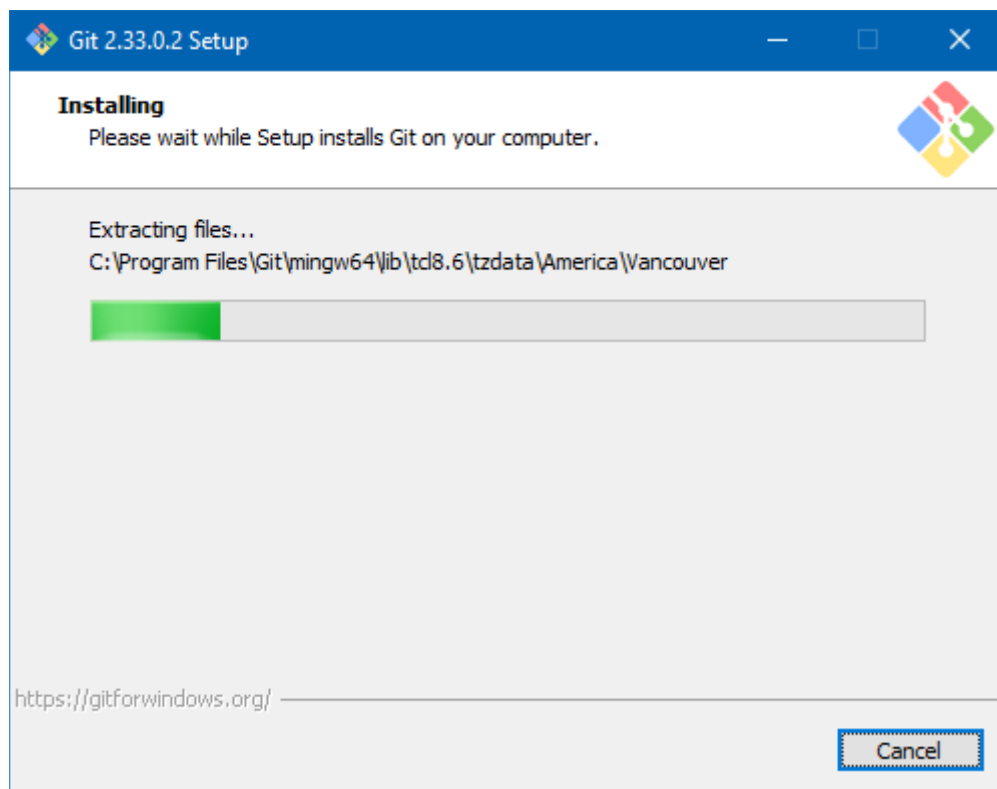




Dejamos todos los valores por defecto e instalamos el programa.







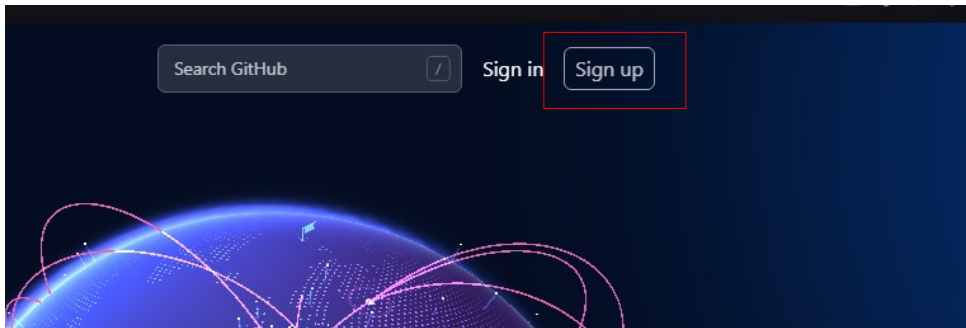
Una vez finalizada la instalación nos dirigimos a la PowerShell para ver si nos reconoce el comando git, para ello ejecutamos `git --version`.

```
PS C:\Users\gonza> git --version
git version 2.33.0.windows.2
PS C:\Users\gonza>
```

Como vemos el programa está instalado correctamente.

2.Crear usuario y repositorio en Github.

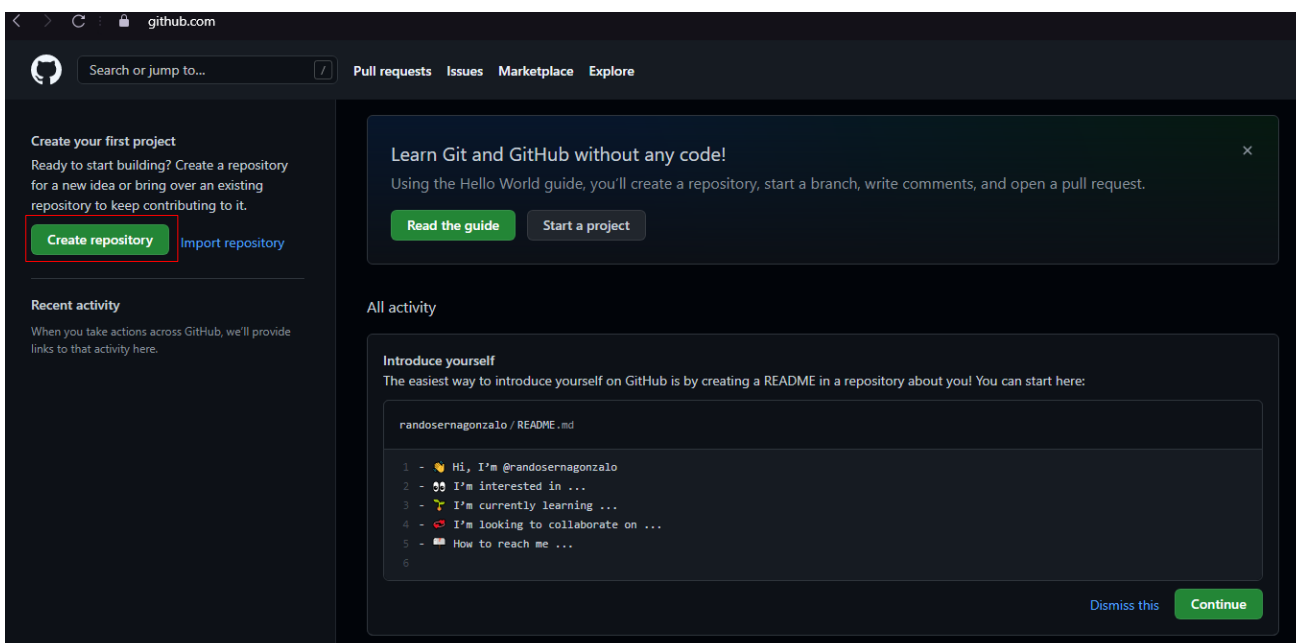
Debemos ir a la página oficial de Github, una vez aquí debemos pulsar el botón de sing up.



Al pulsar este botón debemos escribir nuestro correo electrónico.



Seguidamente nos pedirá una contraseña, una vez proporcionada debemos hacer el captcha y verificar el correo, seguimos las indicaciones que nos proporcione la página y llegaremos al siguiente menú.



En este menú debemos pulsar el botón de create repository y rellenar el campo de nombre con el nombre que queramos que tenga nuestro repositorio.

Create a new repository

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

Owner * randosernagonzalo / Repository name * Proyecto01 ✓

Great repository names are short and snappy. Proyecto01 is available. Inspiration? How about [animated-octo-winner](#)?

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:
Skip this step if you're importing an existing repository.

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

☒ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

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

Create repository

Pulsamos en create repository y nos aparecerá la siguiente pantalla en la que veremos algunos comandos que usaremos mas adelante.

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

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/randosernagonzalo/Proyecto01.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 "# Proyecto01" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/randosernagonzalo/Proyecto01.git
git push -u origin main
```

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

```
git remote add origin https://github.com/randosernagonzalo/Proyecto01.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](#)

Ya habríamos creado nuestro repositorio.

3.Como crear un repositorio local y subirlo a Github.

Una vez instalado git y creado nuestro repositorio, aprenderemos como trabajar con ello. Lo primero que debemos hacer es ejecutar una PowerShell y crear el directorio donde trabajaremos:

```
PS E:\Instituto\1ASIR\BDD\Proyectos> mkdir Proyecto01

Directorio: E:\Instituto\1ASIR\BDD\Proyectos

Mode                LastWriteTime         Length Name
----                -
d-----          28/09/2021    18:39             Proyecto01

PS E:\Instituto\1ASIR\BDD\Proyectos> █
```

Ahora debemos ejecutar el comando *git init* que nos creará un repositorio local en el directorio en el que nos situemos.

```
PS E:\Instituto\1ASIR\BDD\Proyectos> cd .\Proyecto01\
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git init
Initialized empty Git repository in E:/Instituto/1ASIR/BDD/Proyectos/Proyecto01/.git/
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01>
```

Añadimos dos carpetas.

```
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> mkdir src

Directorio: E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01

Mode                LastWriteTime         Length Name
----                -
d-----          28/09/2021    19:11             src

PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> mkdir doc

Directorio: E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01

Mode                LastWriteTime         Length Name
----                -
d-----          28/09/2021    19:11             doc

PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01>
```

Una vez tengamos las carpetas vamos a crear un archivo en la carpeta src, para ello yo usaré el editor de textos “nano” que me permite crear el archivo sin necesidad de dejar la consola.

```
Windows PowerShell
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> nano .\src\holamundo.txt
```

```
Windows PowerShell
v5.7-34-g387900675
Hola mundo
```

Añadiremos el fichero a nuestro repositorio con el comando *git add* .

```
Windows PowerShell
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git add .
warning: LF will be replaced by CRLF in src/holamundo.txt.
The file will have its original line endings in your working directory
```

Ahora debemos hacer un commit con el comando *git commit -m “comit1”*.

```
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git commit -m "comit1"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'gonza@DESKTOP-4856PFV.(none)')
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01>
```

Como vemos necesitamos introducir el correo y el nombre para poder realizar la acción así que escribiremos los comandos que nos sugiere el sistema y volveremos a intentar hacer el commit.

```
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git config --global user.email "granser710@g.educaand.es"
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git config --global user.name "randosernagonzalo"
```

```
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git commit -m "comit1"
[main (root-commit) 40546f8] comit1
 1 file changed, 1 insertion(+)
 create mode 100644 src/holamundo.txt
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01>
```

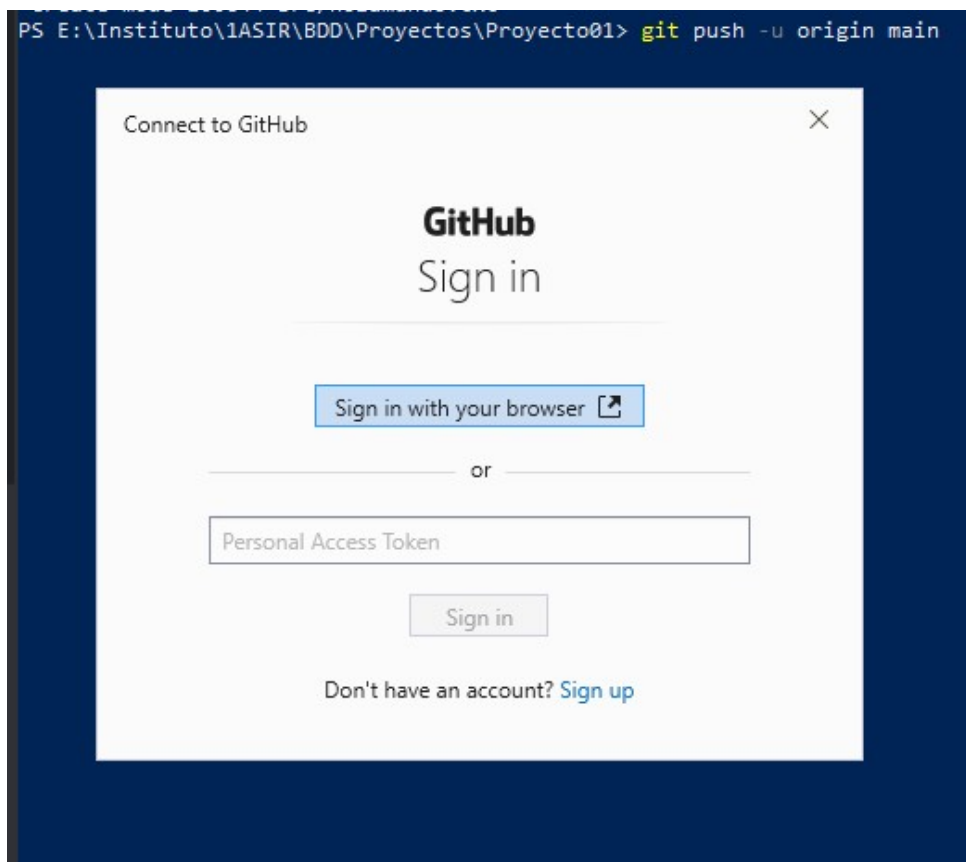
Como vemos, una vez introducido el nombre y el correo el resultado del commit es satisfactorio.

Lo siguiente que haremos sera crear la rama main con el comando *git branch -M main*, si el comando no nos devuelve ningún mensaje es que la operación se ha realizado con éxito.

Debemos darle un origen remoto con el siguiente comando *git remote add origin* <https://github.com/randosernagonzalo/Proyecto01.git> de forma que todos los archivos se suban a mi repositorio en la nube.

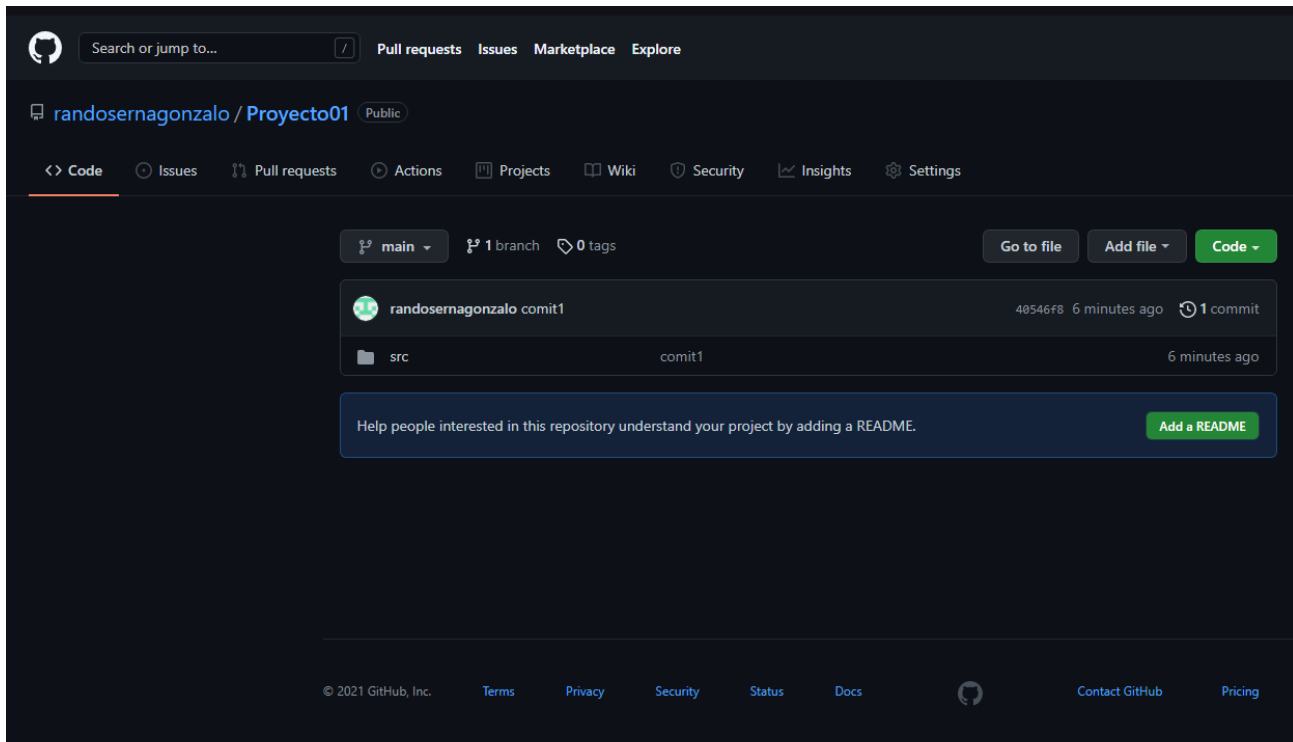
```
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git branch -M main
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git remote add origin https://github.com/randosernagonzalo/Proyecto01.git
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01>
```

Por último ejecutamos el comando *git push -u origin main* para que se nos suba el contenido a la rama main de nuestro repositorio.



Al hacer esto nos aparecerá una pantalla para que nos logueemos y podamos autenticar que somos nosotros los que estamos accediendo.

Una vez autenticado vemos que en nuestro Github ya no salen los comandos, si no que sale nuestro repositorio con el archivo subido.



4. Practicando con algunos comandos.

4.1. Modificación de archivos.

Ahora que ya tenemos nuestro repositorio, vamos a jugar un poco con el, lo primero que haremos es modificar el archivo txt y volverlo a subir al repositorio.

```
Windows PowerShell
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> nano .\src\holamundo.txt
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01>
```

```
Windows PowerShell
v5.7-34-g387900675
Hola mundo, Que tal???? :)_
```

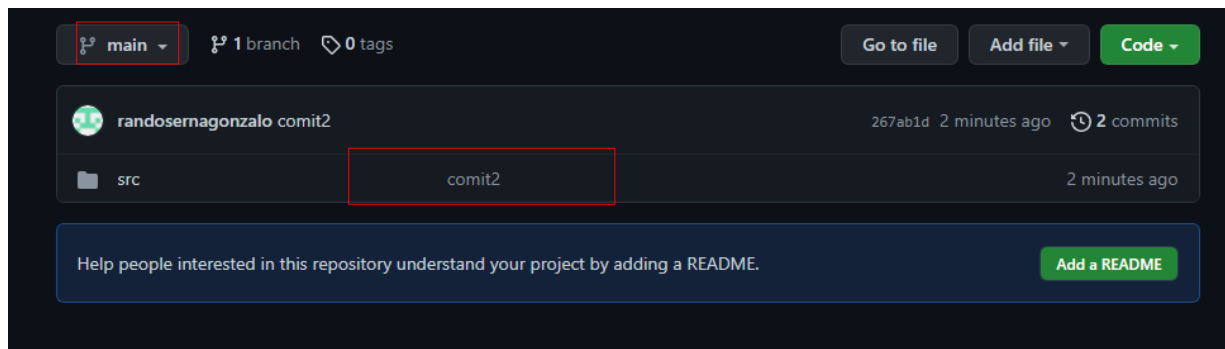
Una vez editado, vamos a hacer el segundo commit.

```
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git add .
warning: LF will be replaced by CRLF in src/holamundo.txt.
The file will have its original line endings in your working directory
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git commit -m "comit2"
[main 267ab1d] comit2
1 file changed, 2 insertions(+), 1 deletion(-)
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01>
```

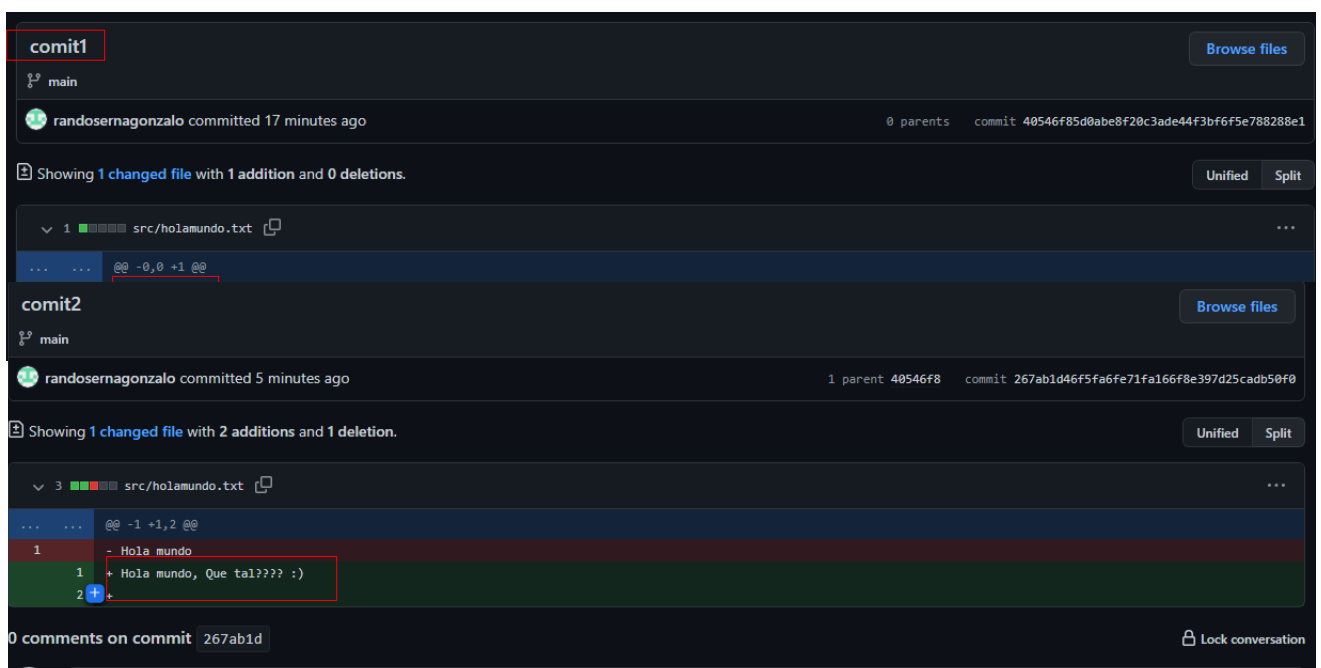
Ahora que tenemos el commit vamos a subirlo a nuestro repositorio.

```
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Writing objects: 100% (4/4), 315 bytes | 315.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/randosernagonzalo/Proyecto01.git
40546f8..267ab1d main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS E:\Instituto\1ASIR\BDD\Proyectos\Proyecto01>
```

Comprobamos que se haya subido correctamente:



Ahora nos dirigimos al comit 1, vemos el archivo y lo comparamos con el del comit 2.



Como vemos los cambios se han realizado con éxito.

4.2. Clonación del repositorio.

Ahora vamos a clonar el repositorio, para ello crearemos una carpeta donde lo clonaremos y ejecutaremos el comando `git clone` <https://github.com/randosernagonzalo/Proyecto01>.

```
Windows PowerShell
PS E:\Instituto\1ASIR\BDD\prueba\p1> git clone https://github.com/randosernagonzalo/Proyecto01
Cloning into 'Proyecto01'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
PS E:\Instituto\1ASIR\BDD\prueba\p1> dir

Directorio: E:\Instituto\1ASIR\BDD\prueba\p1

Mode                LastWriteTime         Length Name
----                -
d-----          28/09/2021   19:43             Proyecto01

PS E:\Instituto\1ASIR\BDD\prueba\p1> █
```

```
PS E:\Instituto\1ASIR\BDD\prueba\p1> dir .\Proyecto01\

Directorio: E:\Instituto\1ASIR\BDD\prueba\p1\Proyecto01

Mode                LastWriteTime         Length Name
----                -
d-----          28/09/2021   19:43             src

PS E:\Instituto\1ASIR\BDD\prueba\p1> nano .\Proyecto01\src\holamundo.txt █
```

```
Windows PowerShell
v5.7-34-g387900675
Hola mundo, Que tal???? :)
█
```

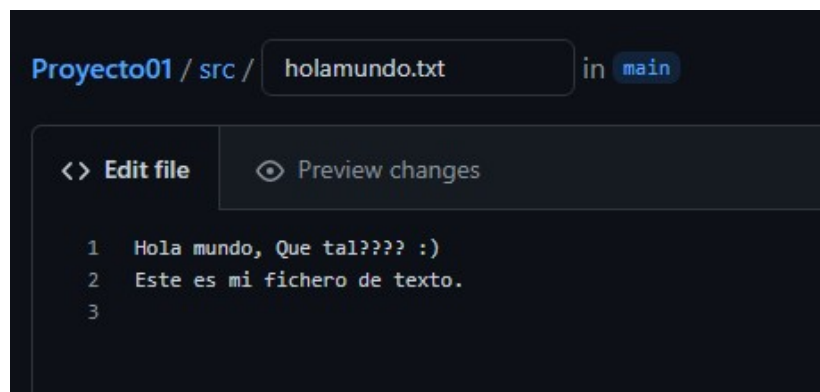
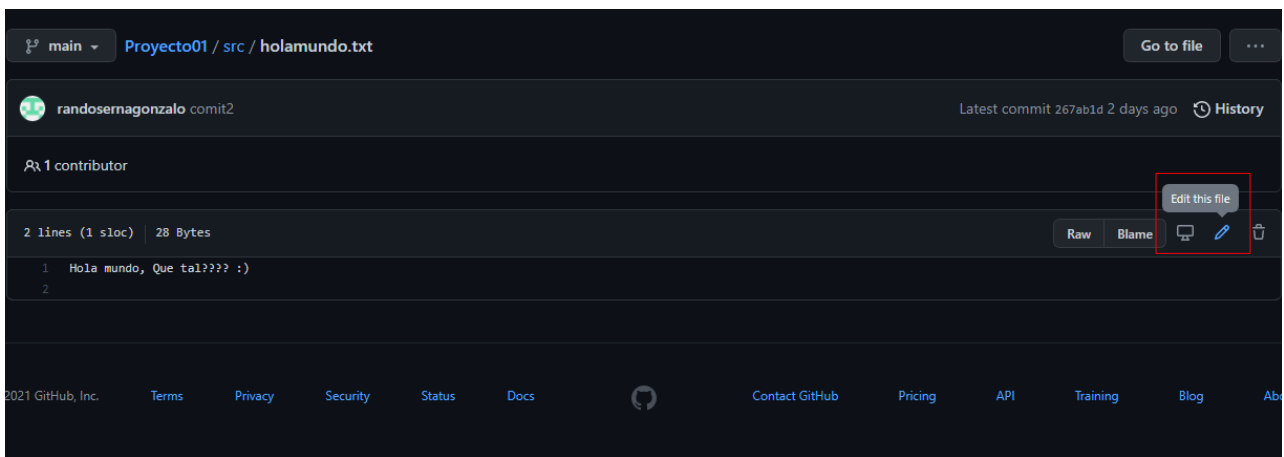
Como vemos hemos clonado el repositorio tal cual lo subimos con el commit mas reciente.

4.3.Actualización local de un repositorio.

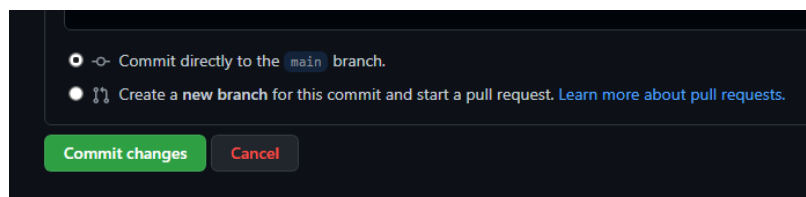
Como hemos visto tenemos repositorios locales y repositorios en la nube, hemos enlazado estos dos de forma que supuestamente deben tener la misma información.

Al igual que podemos editar el contenido de nuestro repositorio localmente para después subirlo a la nube, también podemos editar el contenido directamente desde Github, para bajarnos el contenido editado que no tenemos en nuestro repositorio local usamos el comando *pull*.

Para realizar la prueba nos dirigiremos a nuestro Github y editaremos un poco nuestro fichero txt.



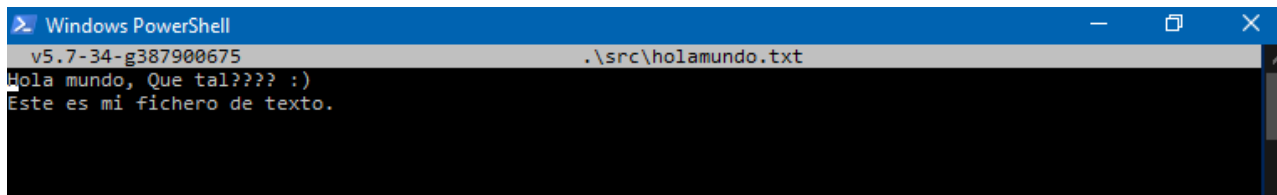
Una vez editado el fichero pulsamos en `commit changes` y nuestro cambio se habrá guardado.



Ahora desde nuestra PowerShell actualizamos nuestro repositorio local ejecutando el comando `git pull` <https://github.com/randosernagonzalo/Proyecto01>.

```
PS E:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto01> git pull https://github.com/randosernagonzalo/Proyecto01
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), 725 bytes | 0 bytes/s, done.
From https://github.com/randosernagonzalo/Proyecto01
 * branch            HEAD       -> FETCH_HEAD
Updating 267ab1d..327c2ed
Fast-forward
 src/holamundo.txt | 2 +--
 1 file changed, 1 insertion(+), 1 deletion(-)
PS E:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto01>
```

Como vemos no nos ha dado ningún tipo de fallo por lo que si ahora al abrir nuestro fichero deberíamos poder ver los cambios realizados.

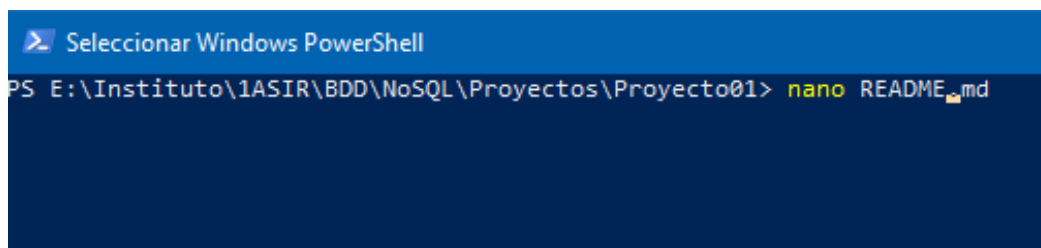


```
Windows PowerShell
v5.7-34-g387900675 .\src\holamundo.txt
Hola mundo, Que tal???? :)
Este es mi fichero de texto.
```

4.4.Creación de readme.

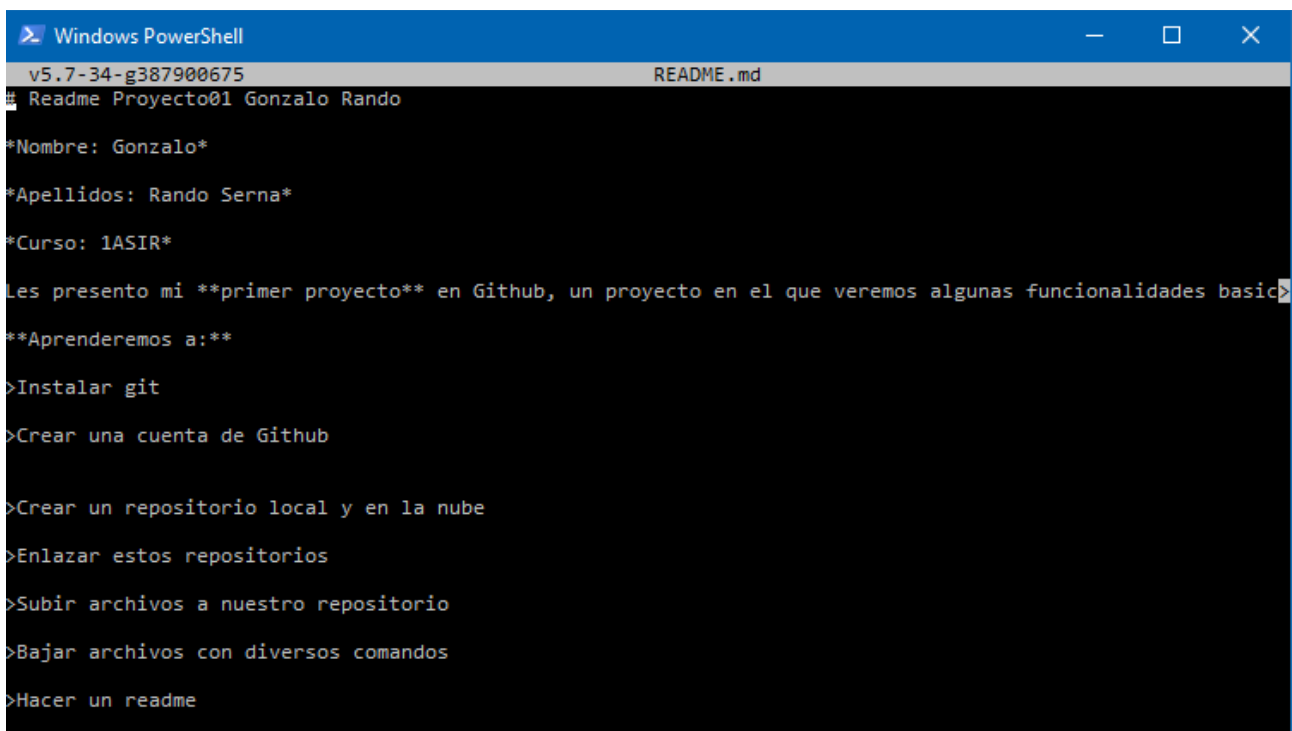
Para un repositorio mas profesional y para dejar nuestra información o datos de interés, vamos a hacer un readme.

Para hacer un readme debemos crear un fichero llamado `readme.md` y escribir en el usando una sintaxis específica.



```
Selección Windows PowerShell
PS E:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto01> nano README.md
```

En el readme mostraré mis datos y algunas explicaciones básicas del proyecto.



```
Windows PowerShell
v5.7-34-g387900675
# Readme Proyecto01 Gonzalo Rando

*Nombre: Gonzalo*

*Apellidos: Rando Serna*

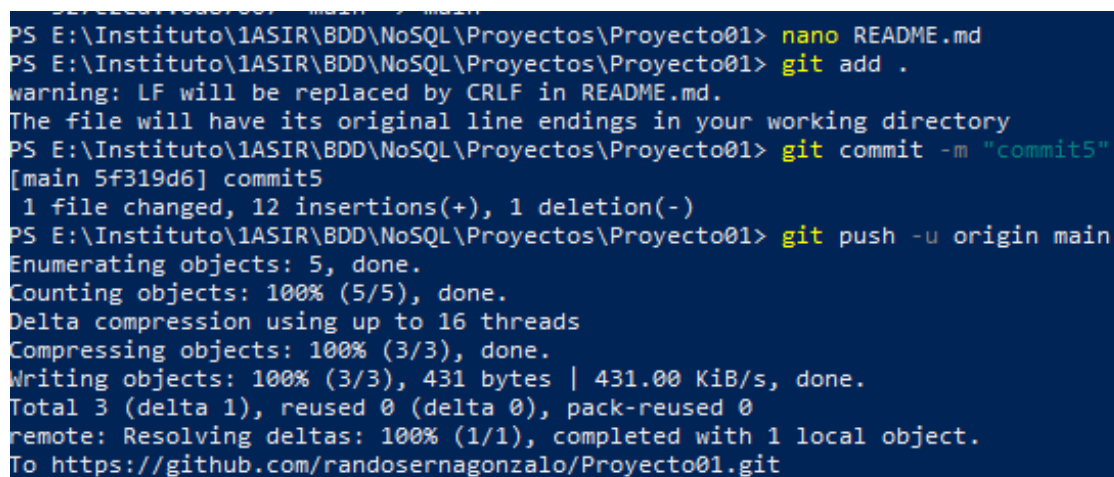
*Curso: 1ASIR*

Les presento mi **primer proyecto** en Github, un proyecto en el que veremos algunas funcionalidades basic

**Aprenderemos a:**

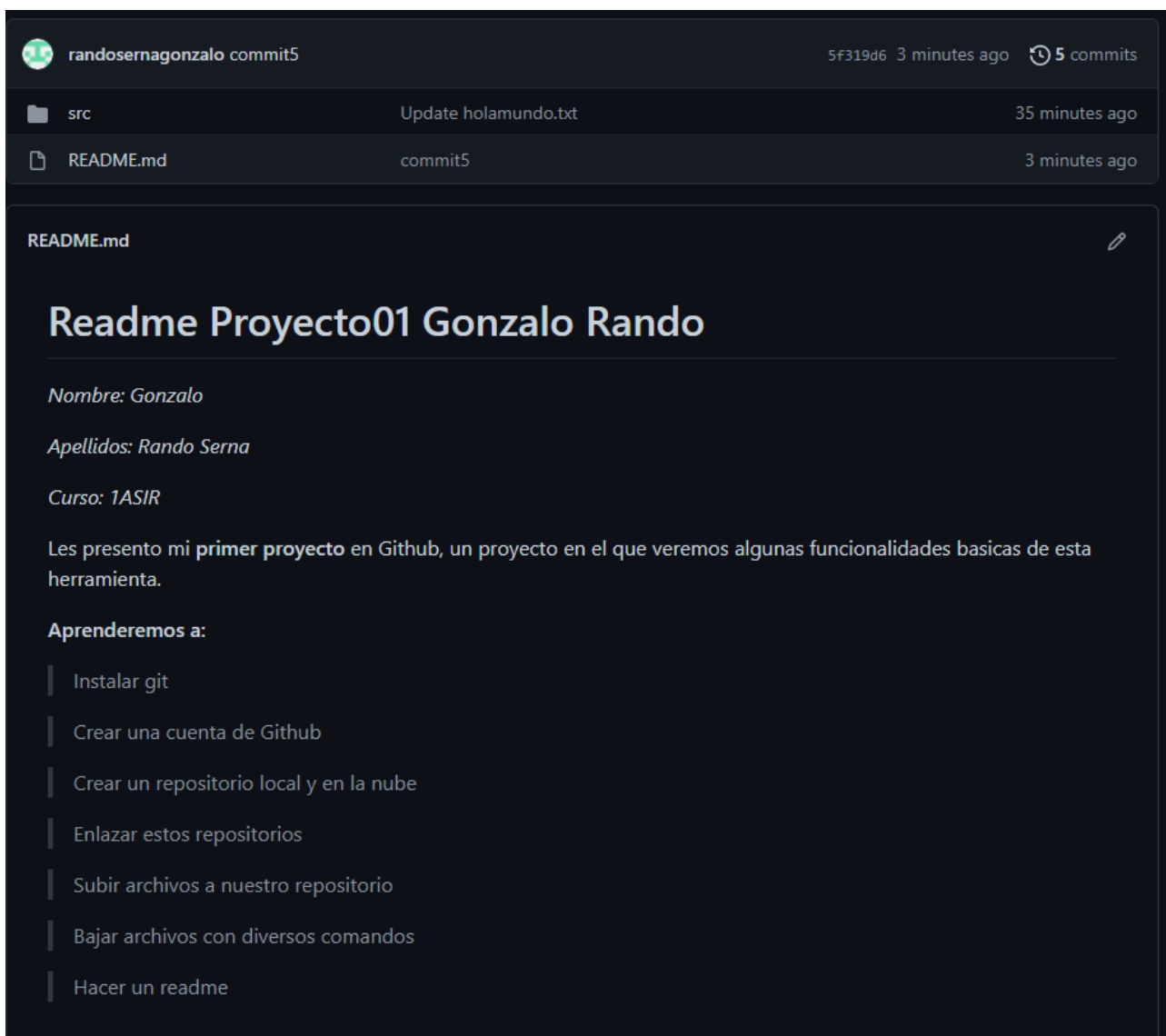
>Instalar git
>Crear una cuenta de Github
>Crear un repositorio local y en la nube
>Enlazar estos repositorios
>Subir archivos a nuestro repositorio
>Bajar archivos con diversos comandos
>Hacer un readme
```

Esta seria la estructura de nuestro readme, ahora tenemos que subirlo a nuestro repositorio y ver que tal se ve en nuestro Github.



```
PS E:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto01> nano README.md
PS E:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto01> git add .
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
PS E:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto01> git commit -m "commit5"
[main 5f319d6] commit5
 1 file changed, 12 insertions(+), 1 deletion(-)
PS E:\Instituto\1ASIR\BDD\NoSQL\Proyectos\Proyecto01> git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 431 bytes | 431.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/randosernagonzalo/Proyecto01.git
```

Accedemos a nuestro Github para comprobar que nuestro readme se ha subido correctamente y se ve como deseamos.



Como vemos nuestro readme se ve bastante bien.