

# *Instalación y configuración de Git y GitHub*



## *Índice:*

1. ¿Qué es GIT? .....	2
2. ¿Qué es GitHub? .....	2
3. Instalación de GIT.....	3
3.1 Descargar instalador GIT. ....	3
3.2 Instalación GIT.....	4
4. Creación de una cuenta en GITHUB. ....	5
5. Creación de nuestro primer proyecto. ....	8
5.1 Creación del árbol del proyecto. ....	8
5.2 Creación del repositorio.....	9
5.3 Vincular los archivos y subirlos al repositorio. ....	10

## 1. ¿Qué es GIT?



Se enfoca en ofrecer un conjunto de herramientas nativas y livianas que traen el conjunto completo de características de Git SCM a Windows al mismo tiempo que brindan interfaces de usuario apropiadas para usuarios experimentados y novatos de Git por igual.

## 2. ¿Qué es GitHub?



Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargar la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo. Como su nombre indica, la web utiliza el sistema de control de versiones.

Un sistema de gestión de versiones es ese con el que los desarrolladores pueden administrar su proyecto, ordenando el código de cada una de las nuevas versiones que sacan de sus aplicaciones para evitar confusiones. Así, al tener copias de cada una de las versiones de su aplicación, no se perderán los estados anteriores cuando se va a actualizar. Permite comparar el código de un archivo para ver las diferencias entre las versiones, restaurar versiones antiguas si algo sale mal, y fusionar los cambios de distintas versiones.

## 3. Instalación de GIT.

### 3.1 Descargar instalador GIT.

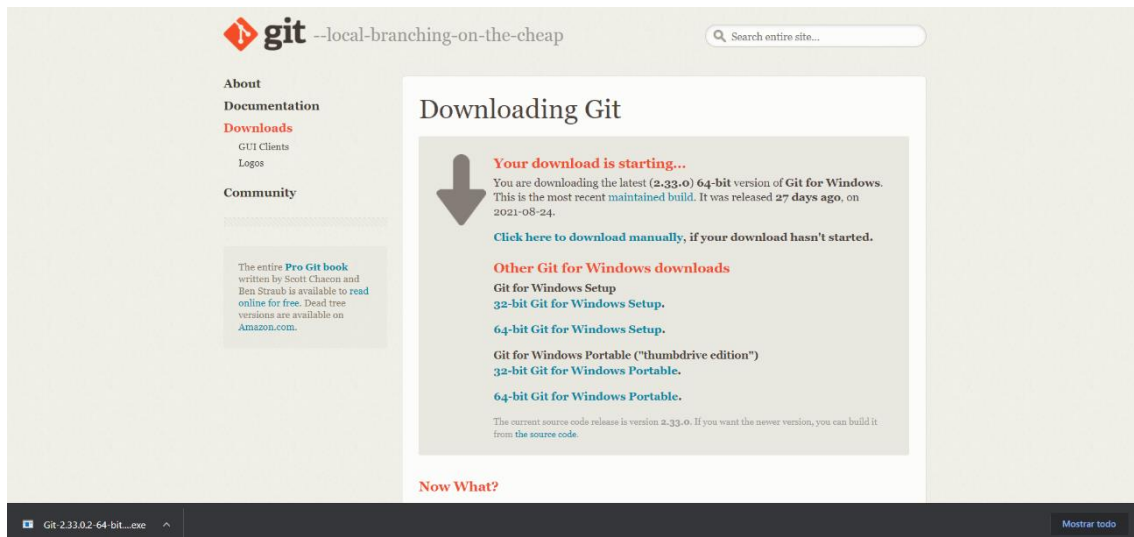
Para comenzar iremos a el siguiente enlace donde podremos descargar el archivo Git.

[git-scm.com](https://git-scm.com)

Al entrar en el enlace nos saldrá la siguiente página:



Podemos ver a la derecha un apartado para descargar la última versión, le clicáremos.



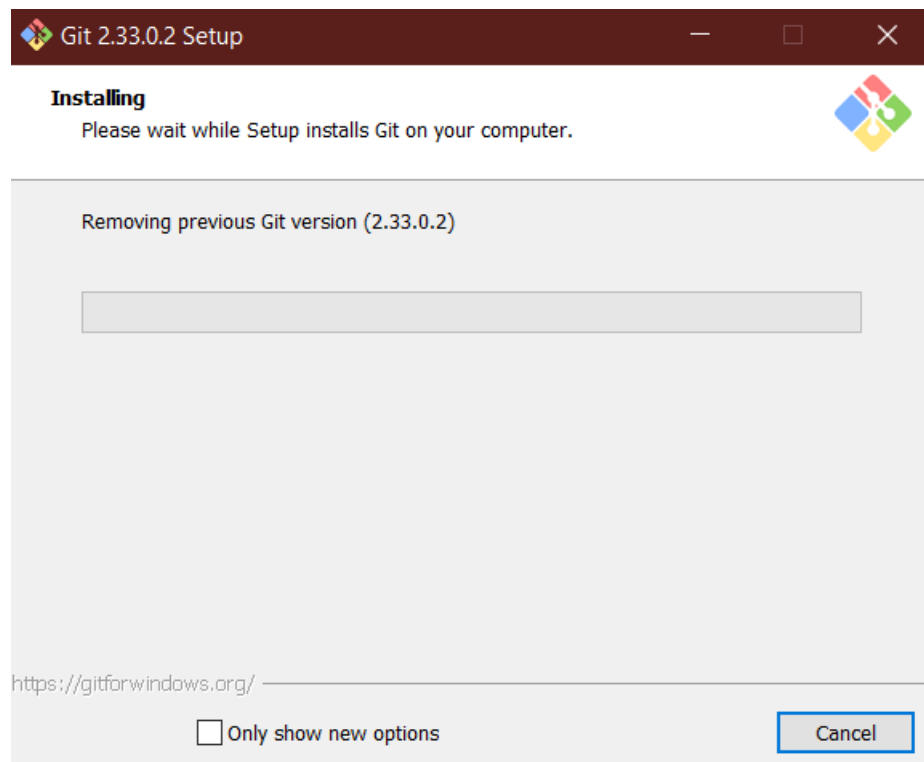
Automáticamente se nos descargara el archivo, pero en caso de lo contrario, podemos seleccionar el tipo de archivo que queremos descargar.

## 3.2 Instalación GIT.

Ejecutamos el archivo de instalación y comenzara la instalación



Para instalarlo le daremos a Next en todas las opciones hasta que empiece a instalar como sale en la siguiente imagen:



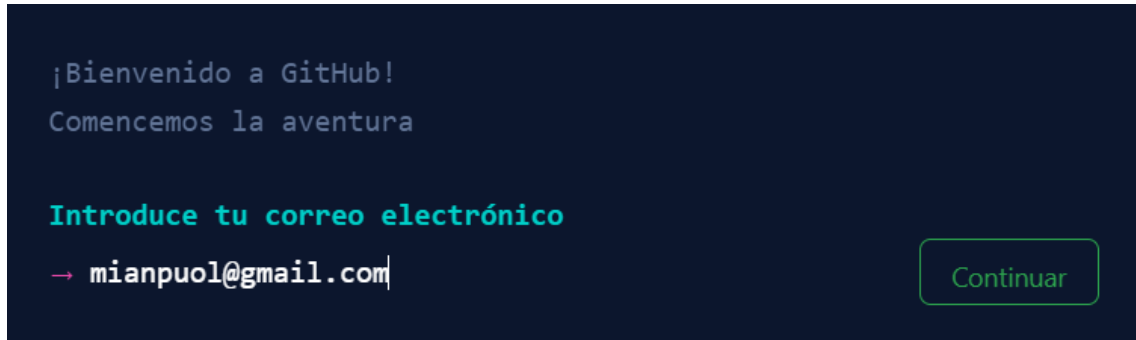
Una vez finalizada la instalación ya estaría Git listo.

## 4. Creación de una cuenta en GITHUB.

En el siguiente paso iremos a crear una cuenta en GitHub, iremos al siguiente enlace:

[github.com](https://github.com)

Dentro iremos a crear cuenta, donde para comenzar, tendremos que introducir un correo, en mi caso usare una cuenta nueva

A dark-themed screenshot of the GitHub sign-up page. At the top, it says '¡Bienvenido a GitHub!' and 'Comencemos la aventura'. Below that, the heading 'Introduce tu correo electrónico' is followed by a text input field containing 'mianpuol@gmail.com' and a green 'Continuar' button on the right.

¡Bienvenido a GitHub!  
Comencemos la aventura

**Introduce tu correo electrónico**

→ mianpuol@gmail.com

Continuar

Luego introduciremos una contraseña.

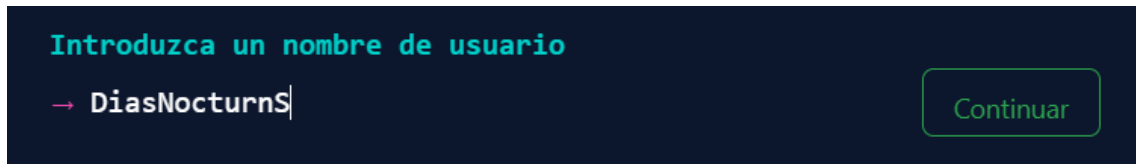
A dark-themed screenshot of the GitHub password creation screen. The heading 'Crea una contraseña' is followed by a text input field with masked dots and a green 'Continuar' button on the right.

**Crea una contraseña**

→ .....|

Continuar

Luego introduciremos un enlace con el que nos loguearemos.

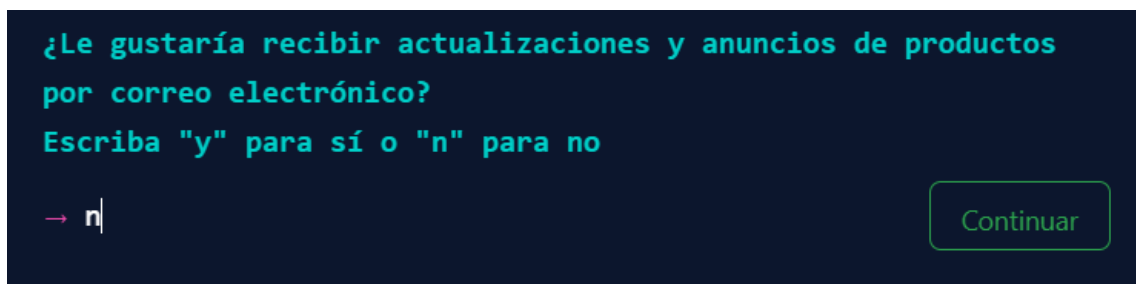
A dark-themed screenshot of the GitHub username screen. The heading 'Introduzca un nombre de usuario' is followed by a text input field containing 'DiasNocturnS' and a green 'Continuar' button on the right.

**Introduzca un nombre de usuario**

→ DiasNocturnS|

Continuar

Nos saldrá si queremos recibir publicidad, en mi caso marco que no.

A dark-themed screenshot of the GitHub marketing preferences screen. It asks '¿Le gustaría recibir actualizaciones y anuncios de productos por correo electrónico?' and 'Escriba "y" para sí o "n" para no'. Below is a text input field with 'n' and a green 'Continuar' button on the right.

**¿Le gustaría recibir actualizaciones y anuncios de productos por correo electrónico?**  
**Escriba "y" para sí o "n" para no**

→ n|

Continuar


Continuamos verificando la cuenta.


### Verifica tu cuenta

**Verificación**

**Solucione este rompecabezas para que sepamos que es una persona real**

Verificar





Nos mandaran un código a nuestro correo el cual introduciremos.

**¡Ya casi terminas!**

Enviamos un código de lanzamiento a **mianpuol@gmail.com**

→ **Introduzca el código**

|

Nos saldrá para cuantos miembros está dedicado y si eres alumno o profesor, en mi caso escojo para mí solo y alumno.

## ¿Cuántos miembros del equipo trabajarán contigo?

Esto nos ayudará a guiarte hacia las herramientas que mejor se adapten a tus proyectos.

Solo yo	2 - 5	5 - 10
10 - 20	20 - 50	50+

## ¿Eres estudiante o profesor?

Estudiante	Maestro
------------	---------

Continuar

Luego sale un apartado el cual no marcaremos nada y le daremos a siguiente.

Nos saldrá una opción en la cual podemos escoger entre pago y gratis, en mi caso escojo gratis.

### Gratis

- ① Repositorios públicos / privados ilimitados
- ① 2,000 acciones minutos / mes  
Gratis para repositorios públicos
- ① 500 MB de almacenamiento de paquetes  
Gratis para repositorios públicos
- ① Soporte comunitario

### Obtenga beneficios adicionales para estudiantes

#### GitHub Pro

- ① **Protege tus ramas**  
Asegúrese de que los colaboradores de su repositorio no puedan realizar cambios irrevocables en las sucursales.
- ① Borradores de solicitudes de extracción
- ① Páginas y wikis
- ① 3000 CI / CD minutos / mes  
Gratis para repositorios públicos
- ① 2GB de almacenamiento de paquetes  
Gratis para repositorios públicos
- ① Soporte basado en web

#### Paquete de desarrollador para estudiantes de GitHub

- ① **Acceso gratuito a las mejores herramientas para desarrolladores de la industria**  
Cientos de ofertas, que incluyen Digital Ocean, Microsoft Azure, Heroku, MongoDB, DataDog, Twilio y Stripe.

Ya tendríamos creada la cuenta en GitHub.



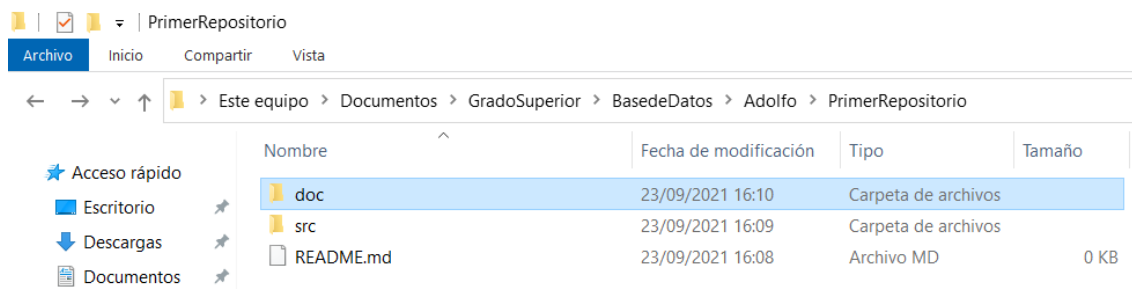
## 5. Creación de nuestro primer proyecto.

### 5.1 Creación del árbol del proyecto.

Para comenzar, crearemos la siguiente estructura, mi carpeta principal la he realizado en:

C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio pero cada uno puede crearla donde quiera.

Como podemos ver contiene 2 carpetas y un archivo README.



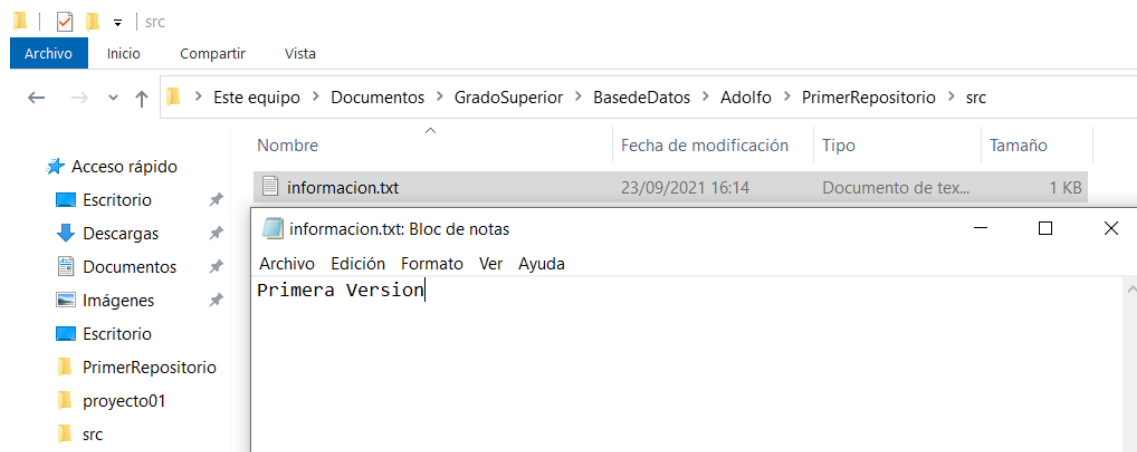
En la carpeta doc tenemos un documento pdf, el cual será el manual de la practica (En el momento de la creación de la practica no esta completado el pdf pero cuando se entregue la tarea estará el definitivo).



Dentro de src tendremos un archivo txt con el que podremos practicar más adelante las modificaciones.

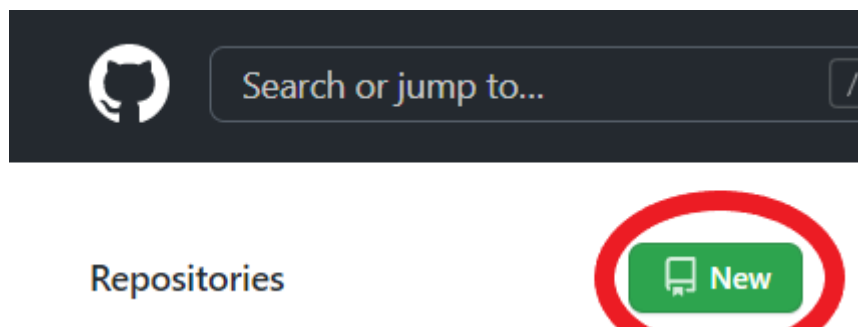


Aquí podemos ver el contenido del archivo.

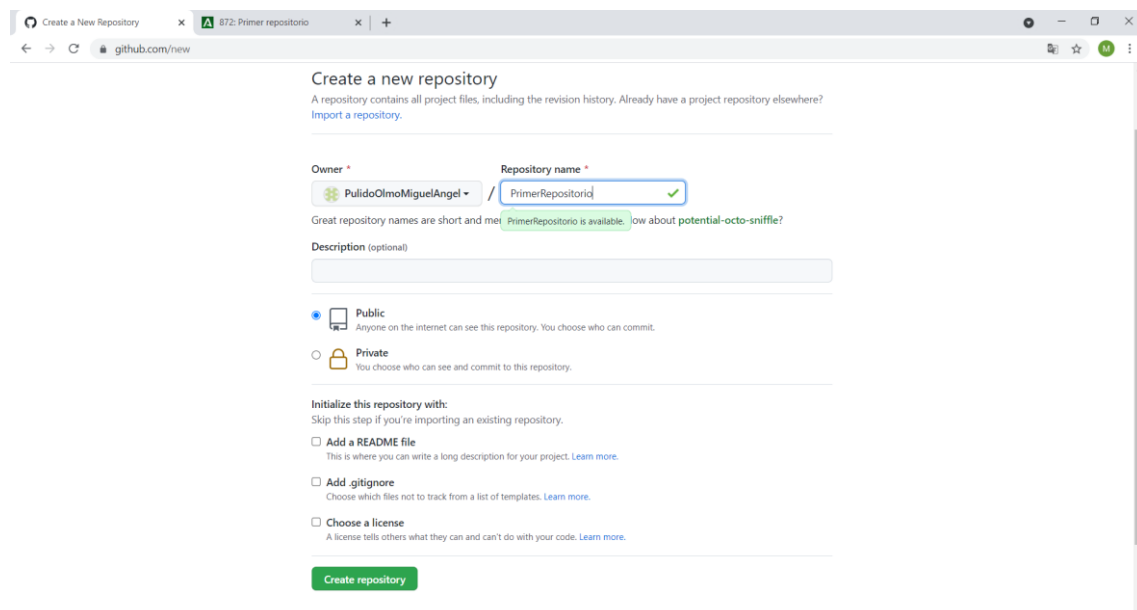


## 5.2 Creación del repositorio.

Nos iremos a GITHUB y crearemos el repositorio.



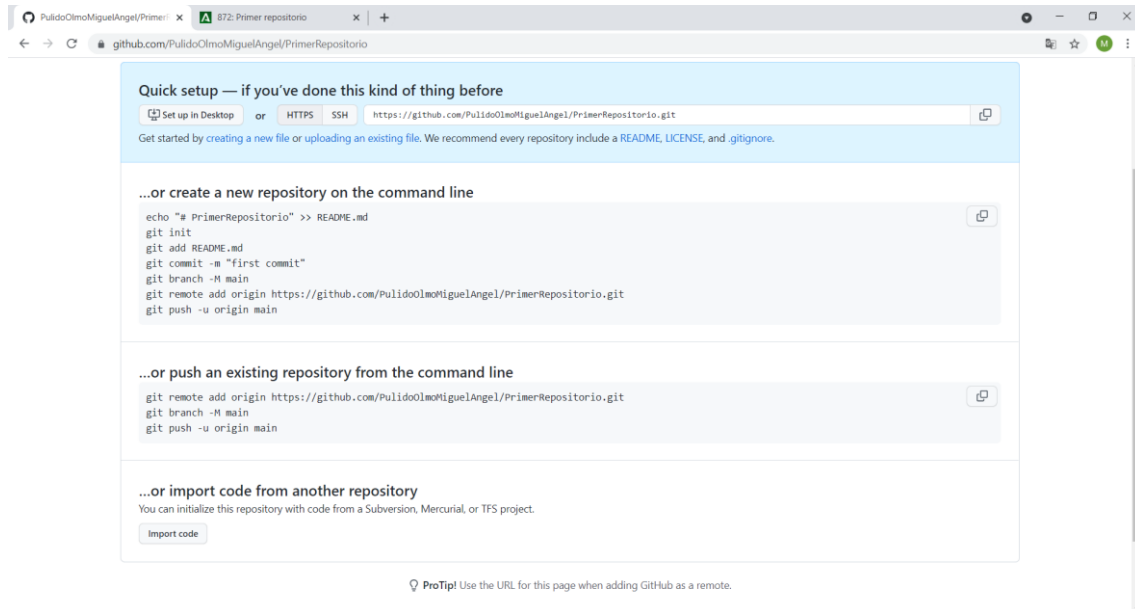
Le pondremos un nombre al repositorio y lo crearemos.



Al crear el repositorio nos saldrá la siguiente página.

Podemos ver que nos da el enlace de nuestro repositorio, esto nos servirá para más adelante.

Debajo podemos ver algunas instrucciones para poder vincular la carpeta del proyecto con el repositorio.



## 5.3 Vincular los archivos y subirlos al repositorio.

Nos iremos al PowerShell y nos colocaremos en la ruta de la carpeta principal.

```
PS C:\Users\alumno> cd C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Iniciaremos el vínculo.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git init
Initialized empty Git repository in C:/Users/alumno/Documents/GradoSuperior/BasedeDatos/Adolfo/PrimerRepositorio/.git/
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Añadiremos los archivos que queremos vincular, subiré el README para que se vea que solo se prepara para la subida el archivo seleccionado y no los otros

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git add README.md
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git status
On branch master

No commits yet

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

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

PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Podemos ver que el archivo README está listo para subirse mientras que el resto no lo está, nosotros queremos subir todo el contenido de la carpeta así que lo haremos con el siguiente comando y podremos ver que están todos preparados.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git add .
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md
    new file:   doc/Proyecto1.pdf
    new file:   src/informacion.txt

PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

A continuación, nos dispondremos a preparar la subida de los archivos añadiéndole una descripción.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git commit -m "Primera Version"
[master (root-commit) 0728ffc] Primera Version
3 files changed, 1 insertion(+)
 create mode 100644 README.md
 create mode 100644 doc/Proyecto1.pdf
 create mode 100644 src/informacion.txt
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Si es la primera vez que subís archivos os pedirá que metáis un correo y un nombre, simplemente le metéis un correo y un nombre

```
*** 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.
```

Dentro del repositorio se puede separar en varias “ramas”, en este paso le diremos a cuál pertenece.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git branch -M main
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Crearemos el vínculo con el enlace que se nos proporcionó anteriormente en la creación del repositorio.

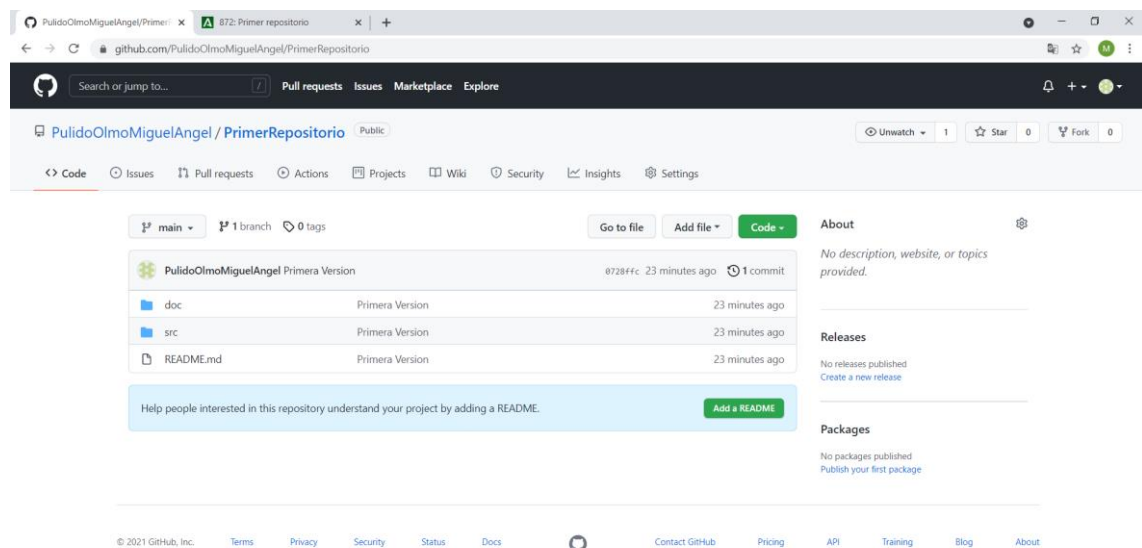
```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git remote add origin https://github.com/PulidoOlmoMiguelAngel/PrimerRepositorio.git
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Miguel Ángel Pulido Olmo

Y por último nos disponemos a subir los archivos.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (7/7), 692.60 KiB | 19.24 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PulidoOlmoMiguelAngel/PrimerRepositorio.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Podemos mirar ya en el repositorio de GitHub como esta subido todo el contenido con el mensaje que le pusimos.



Ahora modificaremos el archivo información.txt que tenemos y lo volveremos a subir.

```
informacion.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Primera Version mas segunda Version.
```

Volvemos a Windows PowerShell y nos iremos a la ruta de la carpeta que contiene el trabajo y repetiremos algunos pasos de los antes dado.

```
PS C:\Users\alumno> cd C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Si le escribimos git status podemos ver que git detecta los archivos que han sido modificados y no están subidos.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/informacion.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Volvemos a preparar el archivo para subirse, le haremos un git add.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git add src/informacion.txt
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/informacion.txt

PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Ahora podemos ver con un status que el archivo está preparado.

Volvemos a prepararle la subida con un mensaje en el que podamos distinguir la segunda versión de la primera.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git commit -m "Segunda Version"
[main 21116e8] Segunda Version
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

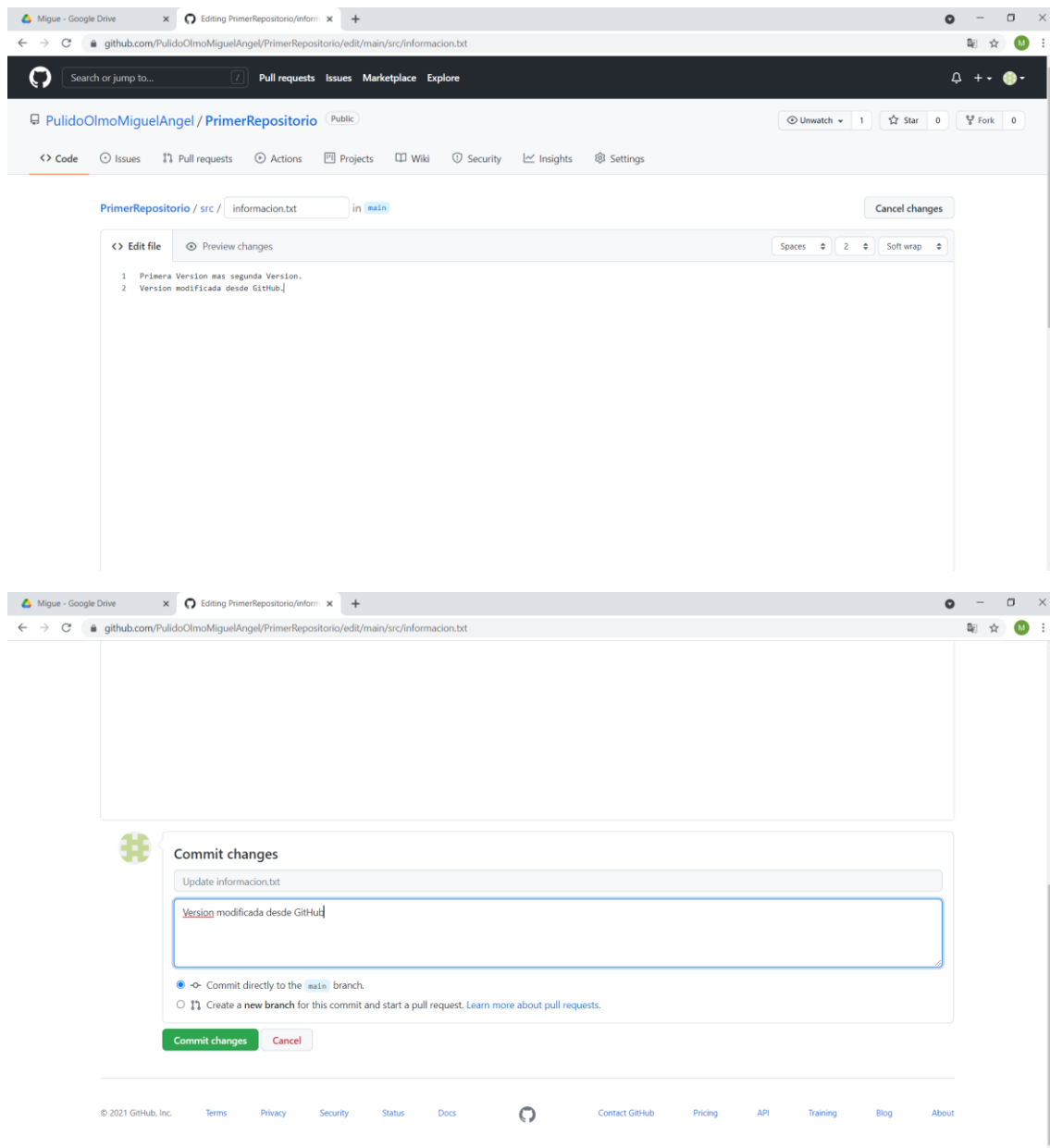
El siguiente paso es subir el archivo a nuestro repositorio:

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 350 bytes | 350.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/PulidoOlmoMiguelAngel/PrimerRepositorio.git
 0728ffc..21116e8  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Podemos ir a GitHub y ver que se ha actualizado la versión del archivo.

PulidoOlmoMiguelAngel Segunda Version		21116e8 21 minutes ago	History
..			
informacion.txt	Segunda Version	21 minutes ago	

Ahora vamos a modificar el archivo desde GitHub y luego nos lo descargaremos en Local.

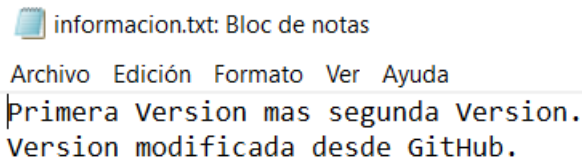


Le daremos a Commit changes y ya estarían guardados los cambios en GitHub.

Ahora descargaremos el archivo para que tengamos el mismo tanto en local como en GitHub.

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio> git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 773 bytes | 33.00 KiB/s, done.
From https://github.com/PulidoOlmoMiguelAngel/PrimerRepositorio
  21116e8..9bb1230  main      -> origin/main
Updating 21116e8..9bb1230
Fast-forward
 src/informacion.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorio>
```

Vamos a comprobar si el archivo se ha actualizado.



Podemos ver que se ha actualizado.

En caso de estar en otro equipo y necesitar descargar el contenido para trabajar desde este, lo podemos hacer de la siguiente manera.

En mi caso he creado una carpeta llamada PrimerRepositorioClone donde meteré la clonación para tener de esta manera la clonación diferenciada de la original (se pueda ver la diferencia).

Nos posicionamos en la localización donde queremos que se descargue, en mi caso será en PrimerRepositorioClone y nos iremos a GitHub al repositorio y copiaremos el enlace del repositorio. Haremos lo siguiente:

```
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorioClone> git clone https://github.com/Pulido01m
oMiguelAngel/PrimerRepositorio
Cloning into 'PrimerRepositorio'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (7/7), done.
Remote: Total 15 (delta 2), reused 10 (delta 1), pack-reused 0 eceiving objects: 46% (7/15)
Receiving objects: 100% (15/15), 693.62 KiB | 3.38 MiB/s, done.
Resolving deltas: 100% (2/2), done.
PS C:\Users\alumno\Documents\GradoSuperior\BasedeDatos\Adolfo\PrimerRepositorioClone>
```

Para finalizar cerraremos sesión para que nadie pueda subir archivos a GitHub.

Iremos al Administrador de credenciales y en el que pone github.com le damos a quitar y ya se eliminaría las credenciales.

