

Registro e instalación de GitHub

+información útil

Contenido

1. ¿Qué es GitHub?	3
2. ¿Cómo nos registramos?	3
3. Instalación de la línea de comandos “git”	6
4. Subir nuestro proyecto	7
5. Como actualizar archivos de nuestro repositorio	12
7. Como descargar nuestro repositorio de GitHub en otro equipo	13
8. Como eliminar las credenciales de GitHub de un sistema	13

1. ¿Qué es GitHub?

GitHub es una herramienta que nos permite conservar nuestros proyectos de manera segura en la nube.

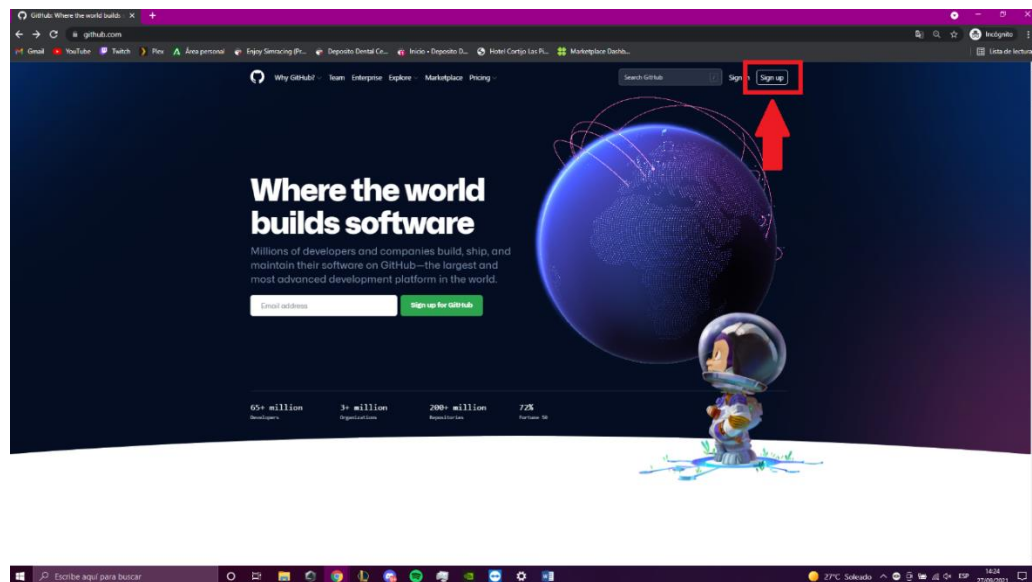
GitHub tiene un sistema de control de versiones que nos permitirá guardar, además de la versión más reciente, todas las versiones anteriores que hayamos subido anteriormente.

2. ¿Cómo nos registramos?

Para guardar nuestros archivos en los servidores de GitHub, deberemos de tener una cuenta creada.

Estos archivos podrán ser vistos por cualquier persona que entre en GitHub, ya que se publican de forma pública. GitHub nos ofrece la posibilidad de poner nuestro proyecto privado, pero es una característica de pago.

Para crear nuestra cuenta iremos a la [página principal de GitHub](https://github.com) y pulsaremos en el botón “Sign up”

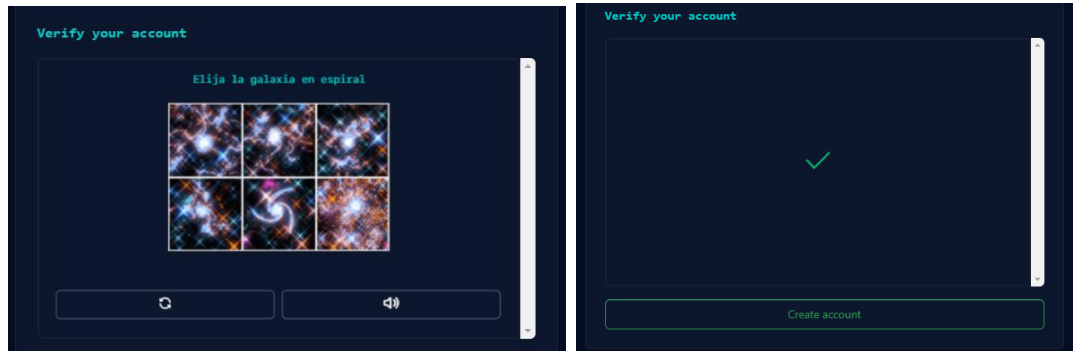


Nos aparecerá la siguiente pantalla, donde tendremos que ir rellenando nuestros datos personales progresivamente:

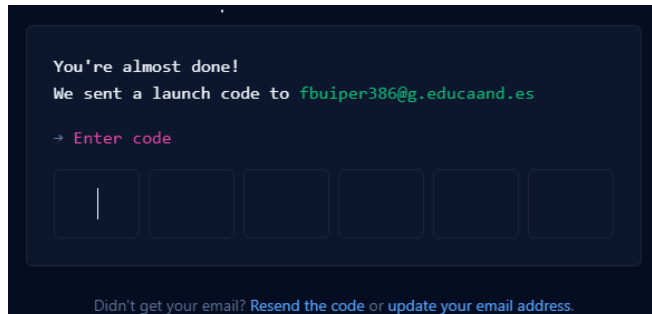
A screenshot of the GitHub registration form. The form is dark-themed and contains the following fields and text: "Welcome to GitHub! Let's begin the adventure". Then, "Enter your email" with a green checkmark and the email "fbuiper386@educaand.es". Next, "Create a password" with a green checkmark and a series of dots. Then, "Enter a username" with a green checkmark and the username "BuizaPerezFrancisco". A green "Continue" button is at the bottom right. At the very bottom, it says "BuizaPerezFrancisco is available."

Registro e instalación de GitHub

Al pulsar “continuar”, tendremos que realizar un “captcha” para que comprueben que no somos un robot:



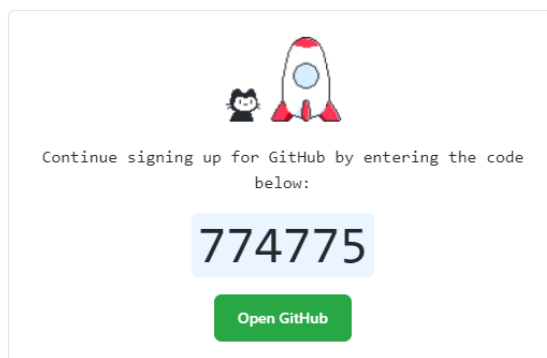
Para verificar el correo electrónico, nos mandarán un código al correo que hemos puesto anteriormente:



Nos vamos al correo e introducimos el código que nos aparece, que será parecido a este:

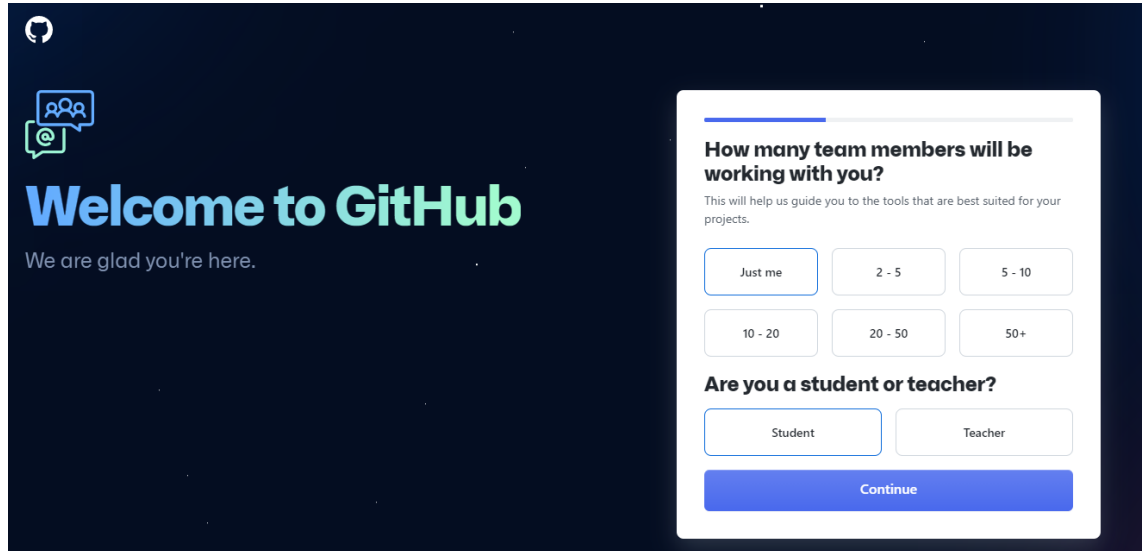


Here's your GitHub launch code,
@BuizaPerezFrancisco!



Registro e instalación de GitHub

Una vez introducido el código, GitHub nos dará la bienvenida

The image shows the GitHub welcome screen. On the left, there's a GitHub logo and a 'Welcome to GitHub' message with the subtitle 'We are glad you're here.' On the right, there's a form titled 'How many team members will be working with you?' with options: 'Just me', '2 - 5', '5 - 10', '10 - 20', '20 - 50', and '50+'. Below this is another section titled 'Are you a student or teacher?' with options: 'Student' and 'Teacher'. At the bottom of the form is a 'Continue' button.

Welcome to GitHub
We are glad you're here.

How many team members will be working with you?
This will help us guide you to the tools that are best suited for your projects.

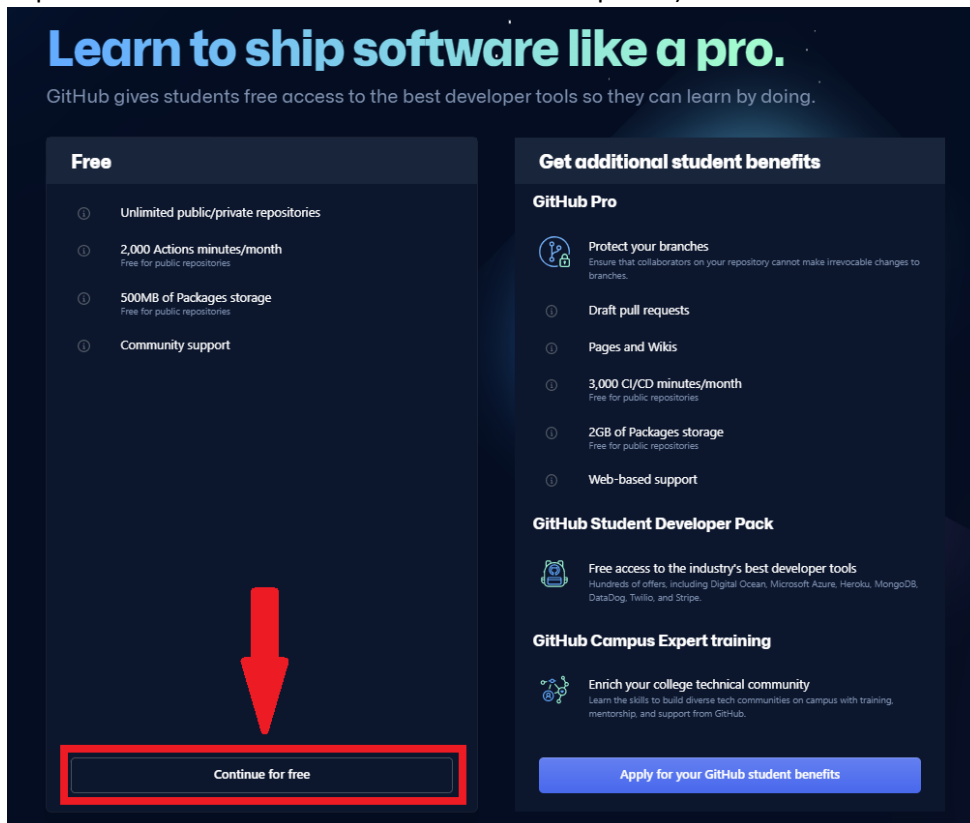
Just me 2 - 5 5 - 10
10 - 20 20 - 50 50+

Are you a student or teacher?

Student Teacher

Continue

Nos preguntará que plan queremos. En mi caso voy a elegir el plan gratuito (ya he explicado anteriormente las diferencias entre los planes):

The image shows the GitHub plan selection screen. It has a dark background with white text. On the left, under the heading 'Free', there's a list of benefits: 'Unlimited public/private repositories', '2,000 Actions minutes/month', '500MB of Packages storage', and 'Community support'. On the right, under the heading 'Get additional student benefits', there's a section for 'GitHub Pro' with benefits like 'Protect your branches', 'Draft pull requests', 'Pages and Wikis', '3,000 CI/CD minutes/month', '2GB of Packages storage', and 'Web-based support'. Below that is the 'GitHub Student Developer Pack' with 'Free access to the industry's best developer tools'. At the bottom right is the 'GitHub Campus Expert training' section. A large red arrow points down to a 'Continue for free' button at the bottom left. At the bottom right is an 'Apply for your GitHub student benefits' button.

Learn to ship software like a pro.
GitHub gives students free access to the best developer tools so they can learn by doing.

Free

- Unlimited public/private repositories
- 2,000 Actions minutes/month
Free for public repositories
- 500MB of Packages storage
Free for public repositories
- Community support

Get additional student benefits

GitHub Pro

- Protect your branches
Ensure that collaborators on your repository cannot make irrevocable changes to branches.
- Draft pull requests
- Pages and Wikis
- 3,000 CI/CD minutes/month
Free for public repositories
- 2GB of Packages storage
Free for public repositories
- Web-based support

GitHub Student Developer Pack

- Free access to the industry's best developer tools
Hundreds of offers, including Digital Ocean, Microsoft Azure, Heroku, MongoDB, DataDog, Twilio, and Stripe.

GitHub Campus Expert training

- Enrich your college technical community
Learn the skills to build diverse tech communities on campus with training, mentorship, and support from GitHub.

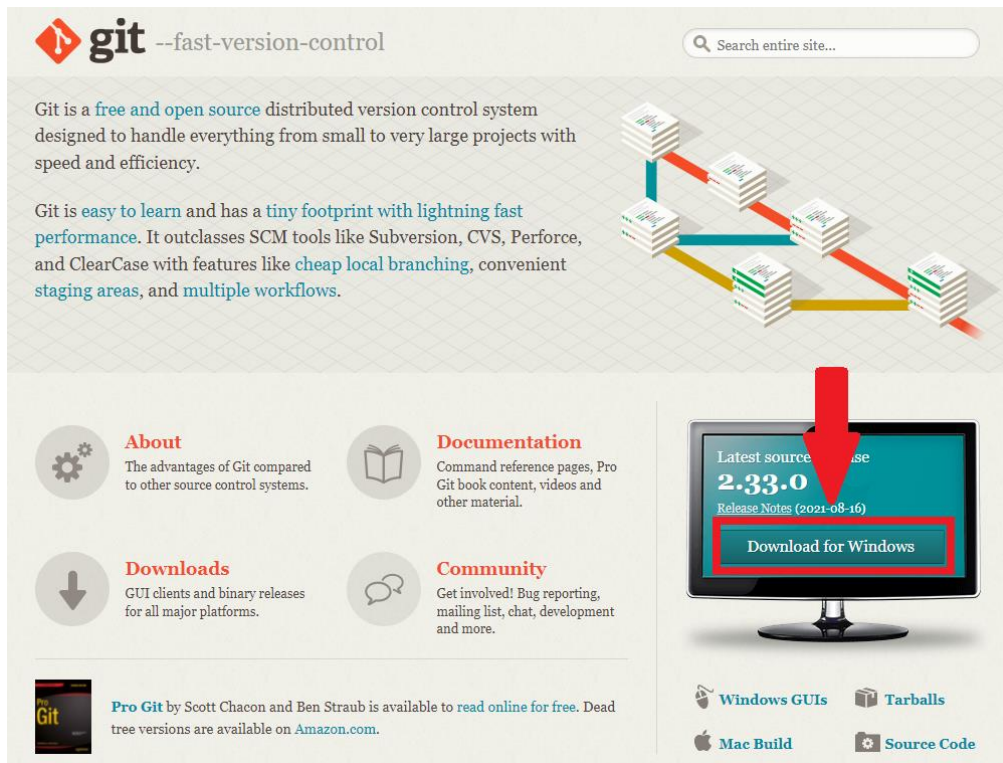
Continue for free

Apply for your GitHub student benefits

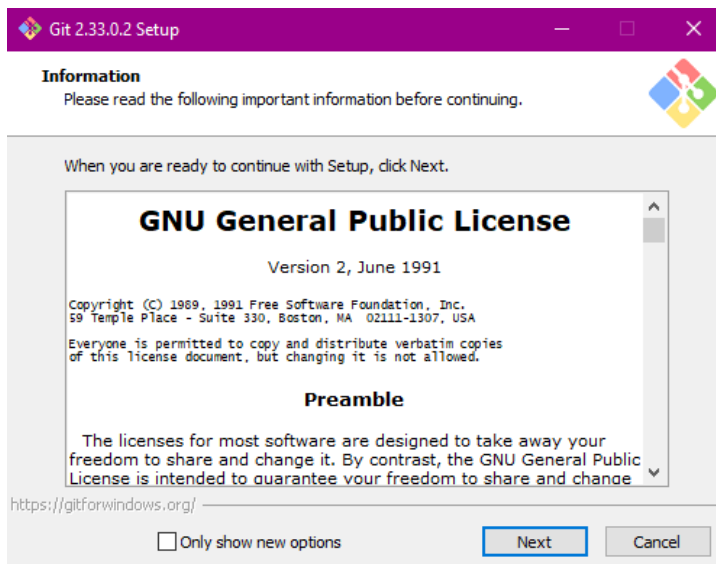
Con todo esto, ya estaríamos totalmente registrados en la página de GitHub.

3. Instalación de la línea de comandos “git”

Para poder usar los comandos de GitHub, deberemos instalar la característica desde la web. Pulsamos en descargar para Windows y ejecutamos el archivo:



Una vez que abramos el archivo, nos aparecerá la siguiente ventana:



Tan solo deberemos continuar todos los pasos (son bastantes ventanas).

Lo dejaremos todo por defecto y procederemos con la instalación.

Ahora podemos probar que se ha instalado correctamente escribiendo en la powershell el comando git --version:

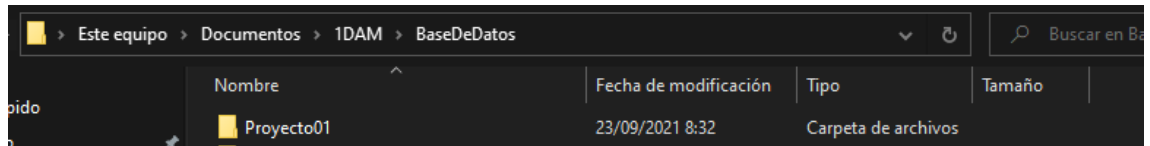
```
PS C:\Users\Francisco> git --version
git version 2.33.0.windows.2
PS C:\Users\Francisco>
```

4. Subir nuestro proyecto

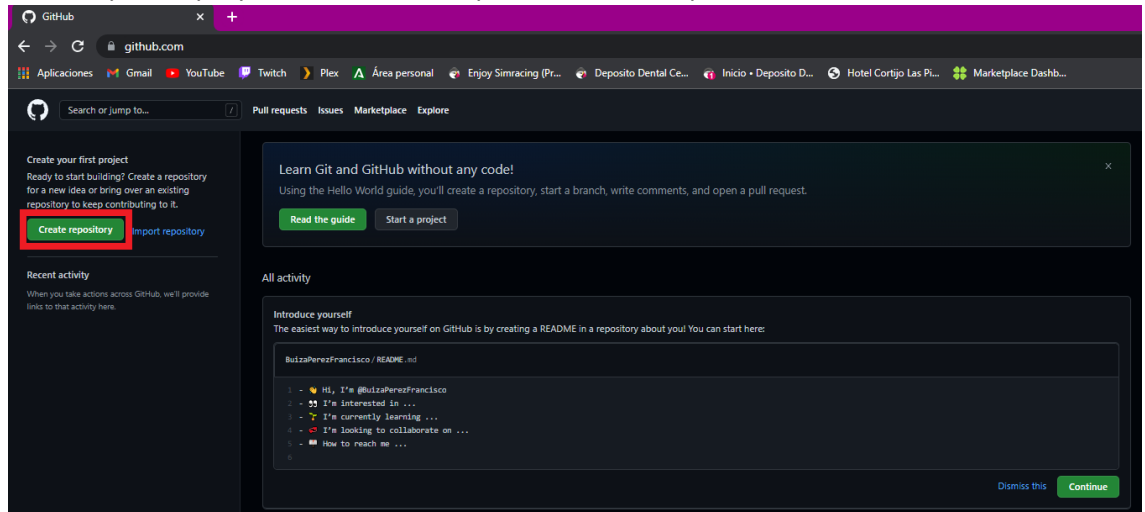
Ahora vamos a preparar todo lo necesario para poder subir archivos al servidor de GitHub:

Primero vamos a crear nuestro repositorio local. Para ello, lo primero que tenemos que hacer es crear un directorio donde vamos a trabajar.

En mi caso voy a seguir la estructura C:\Users\Francisco\Documentos\1DAM\BaseDeDatos, donde voy a crear la carpeta "Proyecto01":

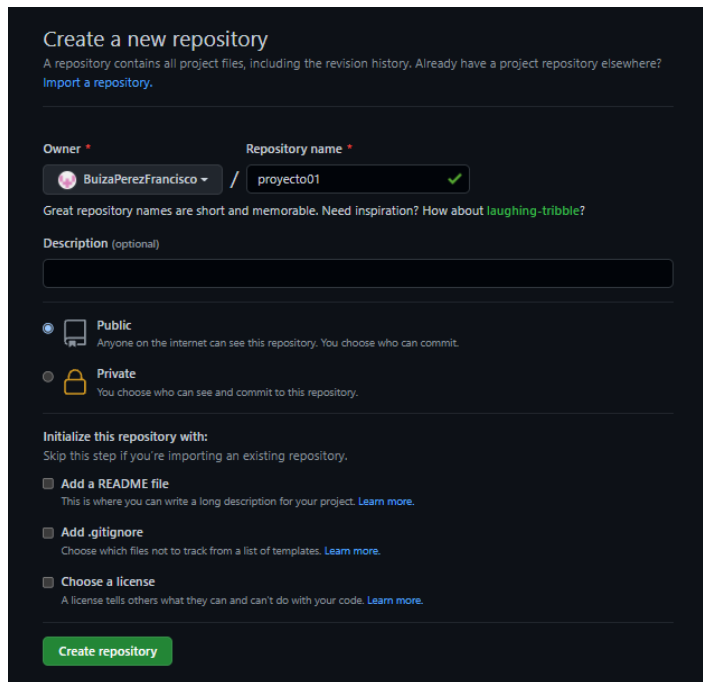


Ahora que hemos creado nuestra carpeta de trabajo, vamos a crear en la página de Git nuestro primer proyecto. Para ello nos pedirá crear un repositorio en la nube:



Al pulsar en el botón de crear repositorio, nos saldrá la siguiente ventana. Tan solo deberemos poner el nombre que queremos y le daremos a crear:

Registro e instalación de GitHub



Create a new repository

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

Owner * Repository name *

BuizaPerezFrancisco / proyecto01

Great repository names are short and memorable. Need inspiration? How about [laughing-tribble?](#)

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.](#)

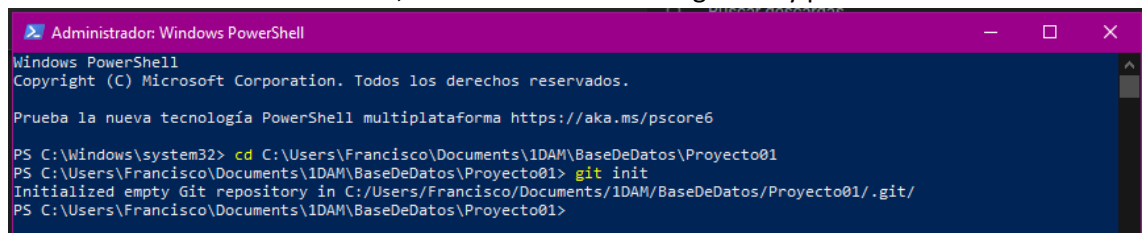
☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

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

Create repository

Una vez creado nuestro repositorio en la nube, vamos a inicializar nuestro repositorio local. Para ello, vamos a abrir la Powershell y nos vamos a situar en el directorio que creamos anteriormente usando el comando “cd”.

Una vez situados en el directorio, usaremos el comando “git init” y pulsaremos intro:



```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Windows\system32> cd C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git init
Initialized empty Git repository in C:/Users/Francisco/Documents/1DAM/BaseDeDatos/Proyecto01/.git/
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

Con esto ya tendríamos inicializado nuestro repositorio local.

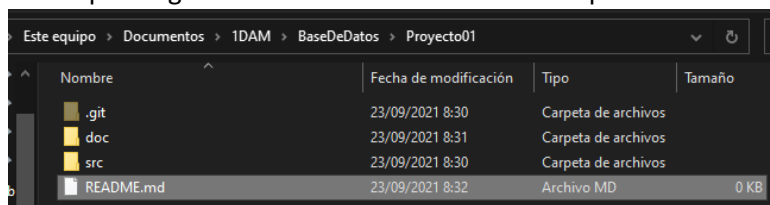
Ahora vamos a crear dentro del repositorio nuestra red de archivos. Yo voy a crear dos carpetas, “doc” y “src”.

IMPORTANTE

Debemos crear un archivo “README.md”

Lo que escribamos en este archivo es lo que la gente verá como información del proyecto en la página de Git.

La carpeta .git es creada automáticamente por Git al inicializar el repositorio.



Nombre	Fecha de modificación	Tipo	Tamaño
.git	23/09/2021 8:30	Carpeta de archivos	
doc	23/09/2021 8:31	Carpeta de archivos	
src	23/09/2021 8:30	Carpeta de archivos	
README.md	23/09/2021 8:32	Archivo MD	0 KB

Podemos darle formato al texto del “README.md”

Podemos ver la plantilla de formato en [esta página](#).

Ya que tenemos todo listo, vamos a ver el “status” de nuestro repositorio con el comando “git status”

Registro e instalación de GitHub

```
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
```

Como podemos ver, nos dice que los archivos en rojo no están incluidos para el próximo empaque.

Para agregarlos, tan solo deberemos escribir el comando “git add .” (en caso de que queramos subir todos los archivos) desde el repositorio. Después podemos comprobar que se han incluido con “git status”:

```
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git add .
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> 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/Instrucciones.txt
        new file:   src/programaa.txt

PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

Para subir nuestro repositorio, primero tenemos que empaquetar los archivos. Para ello, usaremos el comando “git commit -m” y el nombre con el que queremos llamar al envío. Como podemos ver, nos pide que pongamos nuestra identidad. Introducimos los datos que nos pide, con los comandos rodeados por el cuadrado rojo:

```
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git commit -m "Primer envio"
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 'Francisco@portatil1.(none)')
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git config --global user.email "fbuiper386@educaand.es"
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git config --global user.name "Francisco Buiza"
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git commit -m "Primer envio"
[master (root-commit) 5c594fc] Primer envio
 3 files changed, 4 insertions(+)
 create mode 100644 README.md
 create mode 100644 doc/Instrucciones.txt
 create mode 100644 src/programaa.txt
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

Ahora tenemos que dirigirnos a la rama principal del proyecto. Para ello usaremos el comando “git branch -M main”:

```
Administrador: Windows PowerShell
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git branch -M main
```

Ahora vamos a vincular nuestro repositorio local con el repositorio de la nube. Para ello, iremos a la página de GitHub y buscaremos al URL de nuestro proyecto:

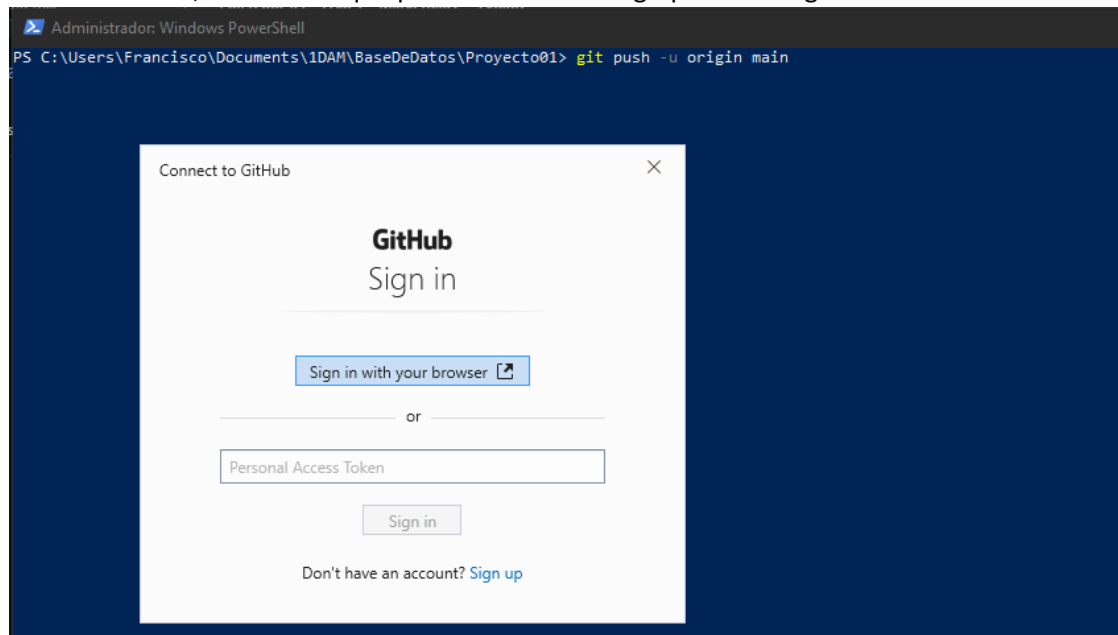


La copiamos y lo escribimos en la powershell con el comando “git remote add origin”:

```
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git remote add origin https://github.com/BuizaPerezFrancisco/proyecto01.git
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

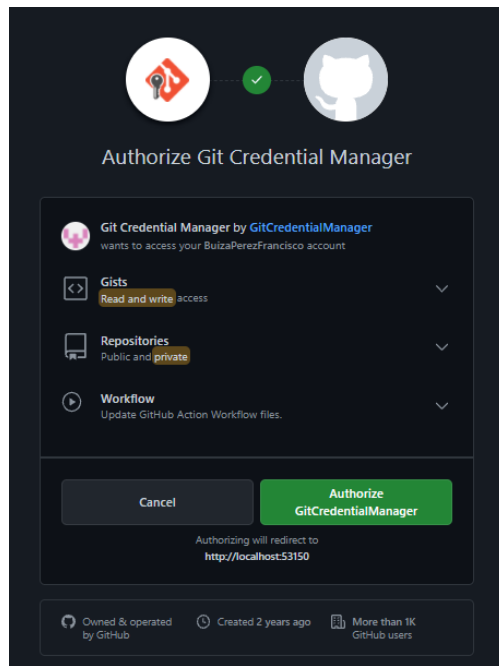
Ahora vamos a intentar subir los archivos al repositorio en la nube pero, como podremos comprobar ahora, nos pedirá autenticarnos por ser la primera vez que lo usamos y que de esa forma pueda confirmar los datos que le pasamos anteriormente.

Para hacer esto, tendremos que poner el comando “git push -u origin main”:

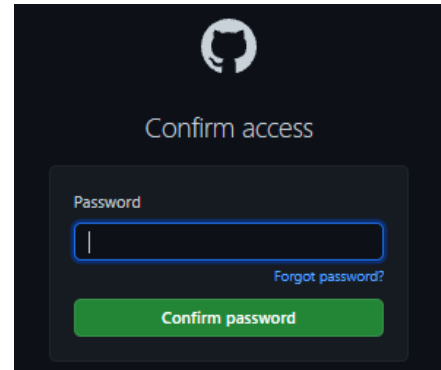


Le damos al botón de “Sign in with your browser” y nos mandará hacia nuestro navegador predeterminado:

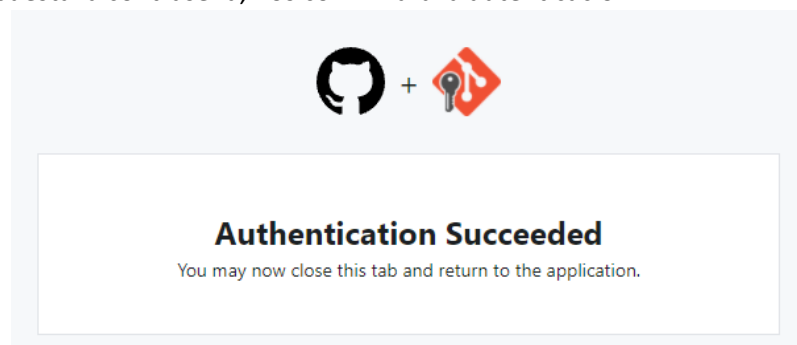
Registro e instalación de GitHub



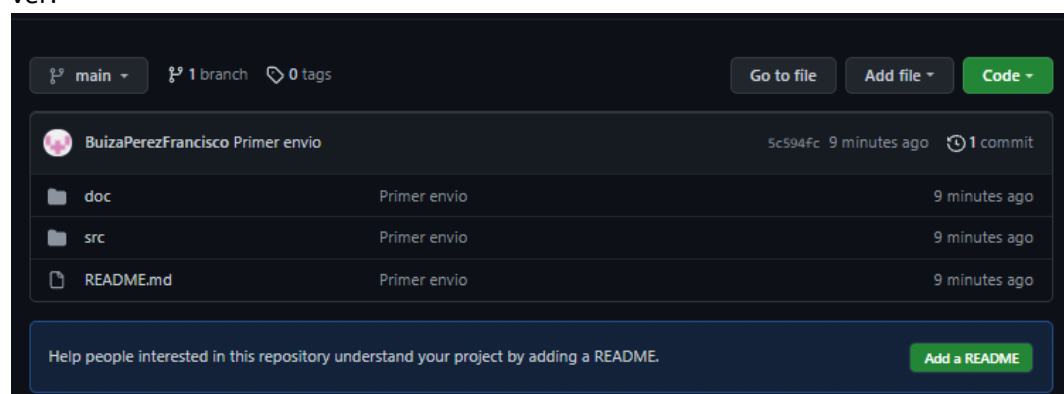
Aquí autorizamos el acceso a nuestra cuenta de GitHub (no hará falta iniciar la sesión si ya la tenemos iniciada en el navegador predeterminado, aunque tendremos que volver a poner la contraseña por seguridad:



Una vez puesta la contraseña, nos confirmará la autenticación:

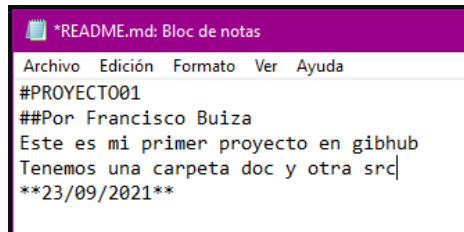


Una vez terminada la autenticación, los archivos ya se habrán subido como podemos ver:



5. Como actualizar archivos de nuestro repositorio

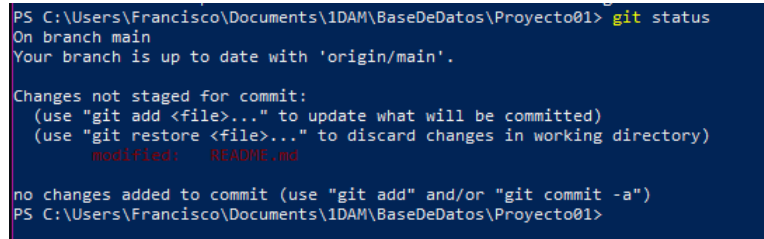
Para hacer esto, vamos a modificar el archivo README.md con el siguiente texto:



```
*README.md: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
#PROYECTO01
##Por Francisco Buiza
Este es mi primer proyecto en gihub
Tenemos una carpeta doc y otra src
**23/09/2021**
```

Esto dará información a los usuarios que entren a nuestro proyecto.

Con “git status” podemos comprobar si un archivo ha cambiado:

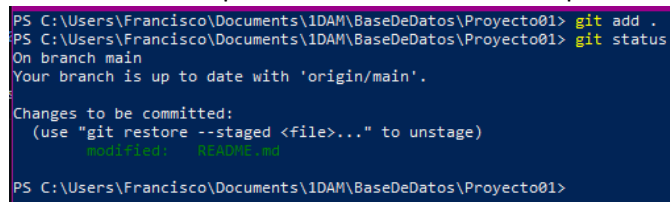


```
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git status
On branch main
Your branch is up to date with 'origin/main'.

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:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

Vamos a añadirlo para tenerlo en cuenta en el próximo empaque:

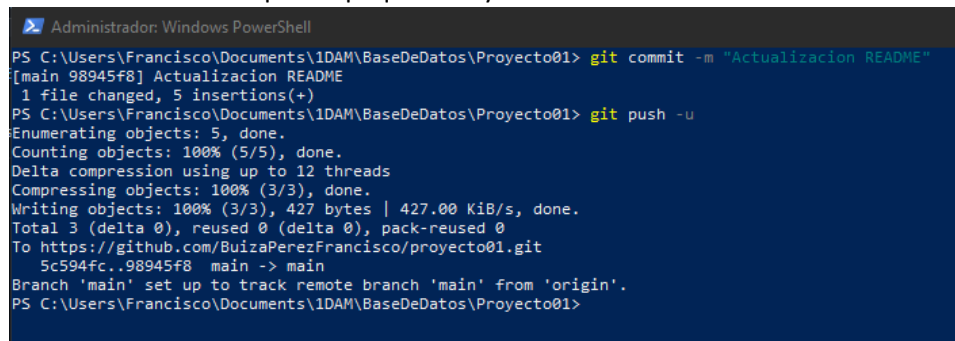


```
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git add .
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

Ahora solo tenemos que empaquetarlo y subirlo como hemos visto anteriormente:

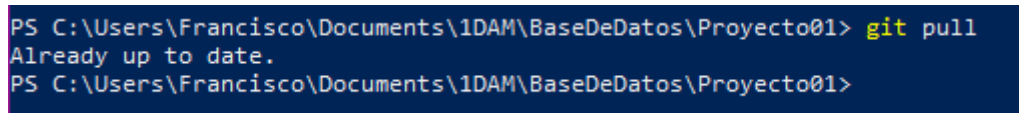


```
Administrador: Windows PowerShell
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git commit -m "Actualizacion README"
[main 98945f8] Actualizacion README
 1 file changed, 5 insertions(+)
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git push -u
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 427 bytes | 427.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/BuizaPerezFrancisco/proyecto01.git
   5c594fc..98945f8  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

6. Como actualizar nuestro repositorio local si hemos modificado algo desde GitHub

Para ello tan solo deberemos escribir el comando “git pull”

Con ese comando, nuestro repositorio local se sincronizaría totalmente con el de GitHub.

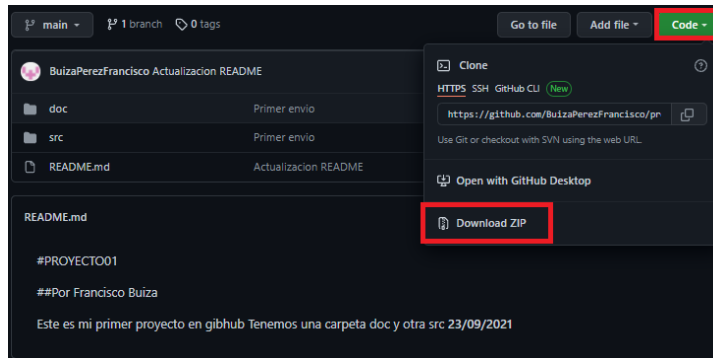


```
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git pull
Already up to date.
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

En este caso estaba todo actualizado.

7. Como descargar nuestro repositorio de GitHub en otro equipo

Tenemos varias formas de hacerlo. Una de ellas es descargar directamente los archivos en .ZIP



Otra manera es instalando de nuevo los comandos de GitHub en el otro equipo y escribiendo el comando que nos descarga el repositorio completo.

Tan solo deberemos escribir el comando “git pull” seguido de la URL del repositorio que queremos descargar (debemos estar situados con la powershell en el directorio en que queremos descargar el repositorio).

```
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01> git clone https://github.com/BuizaPerezFrancisco/proyecto01
Cloning into 'proyecto01'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 13 (delta 0), reused 13 (delta 0), pack-reused 0
Receiving objects: 100% (13/13), done.
PS C:\Users\Francisco\Documents\1DAM\BaseDeDatos\Proyecto01>
```

8. Como eliminar las credenciales de GitHub de un sistema

Para eliminar los permisos de cambiar nuestros repositorios desde un ordenador (ya sea porque hemos usado un ordenador que no sea nuestro o cualquier otro motivo), debemos ir al apartado de Windows de “Administrador de credenciales” en el Panel de Control.

Una vez dentro, buscaremos el apartado de “github.com” y quitaremos las credenciales:

