



REPOSITORIO GITHUB

1ºDAM GESTIÓN BASE DE DATOS

Consolación Begines Roldán

ÍNDICE

1. ¿QUÉ ES Y PARA QUÉ SIRVE GITHUB?.....	1
1.1. ¿Qué es Git?.....	1
1.2. ¿Qué es un control de versión?.....	1
2. COMENZAR CON GITHUB.....	2
2.1. Descargar GitHub.....	2
2.2. Instalación.....	2
2.3. Creación y configuración de la cuenta GitHub.....	2
3. CREACIÓN DE UN NUEVO REPOSITORIO.....	3
3.1. Paso 1.....	4
3.2. Paso 2.....	4
3.3. Paso 3.....	4
3.4. Paso 4.....	4
3.5. Paso 5.....	4
4. WINDOWS POWERSHELL.....	5
4.1. Abrir programa.....	5
4.2. Comando cd.....	5
4.3. Comando git init.....	6
4.4. Comando git status.....	6
4.5. Comando git add	7
4.6. Comando git commit -m.....	7
4.7. Comando git config --global user,email y comando git config --global user.name.....	8
4.8. Comando git branch -M main.....	8
4.9. Comando git remote add origin.....	9
4.10. Comando git push -u origin main.....	10
5. CONEXIÓN DE CUENTAS Y FINALIZACIÓN DEL PROYECTO.....	10

REPOSITORIO GITHUB

1 ¿Qué es y para qué sirve GitHub?

Github es un repositorio online gratuito que se utiliza para gestionar proyectos y controlar versiones de código, permitiendo a los desarrolladores administrar cambios en un software a la vez que el proyecto evoluciona. Es muy utilizado por desarrolladores para almacenar sus trabajos dando así la oportunidad a millones de personas de todo el mundo a cooperar en ellos.



Se podría hablar de Github como la red social pensada para desarrolladores, siendo este repositorio uno de los más usados a nivel mundial.

Podemos seguir e interactuar con personas interesadas en un tipo de proyecto en concreto, dando a conocer los nuestros o cooperando en el proyecto de terceros.

Para entender exactamente qué es GitHub, primero se necesita conocer los dos principios que lo conectan:

- Git
- Control de versión

1.1 ¿Qué es Git?

Git es un sistema de control de versiones distribuido de código abierto desarrollado por Linus Torvalds, el creador de Linux.

Un sistema de control de versión, lo que quiere decir que la base del código entero y su historial se encuentran disponibles en la computadora de todo desarrollador, así permite un fácil acceso a las bifurcaciones y fusiones.

1.2 ¿Qué es un Control de versión?

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Usar un control de versiones, también significa generalmente que si arruinas o pierdes archivos, será posible recuperarlos fácilmente.

2 Comenzar con GitHub.

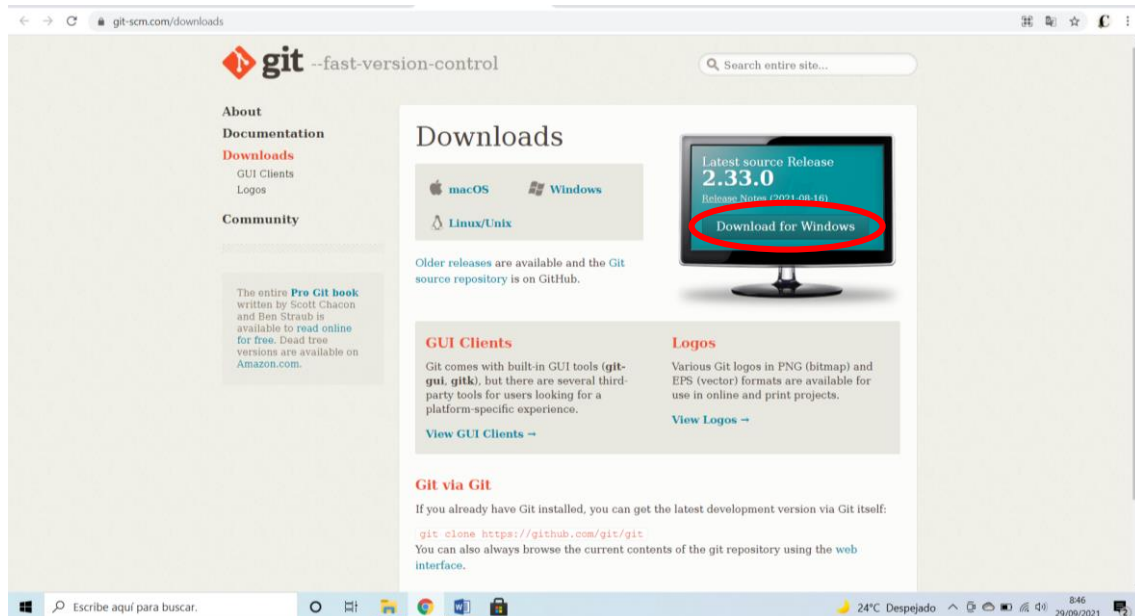
Para llevar a cabo nuestro proyecto, debemos seguir unos pasos:

2.1 Descargar GitHub

Podemos descargar GitHub fácilmente mediante el enlace:

<https://git-scm.com/downloads>

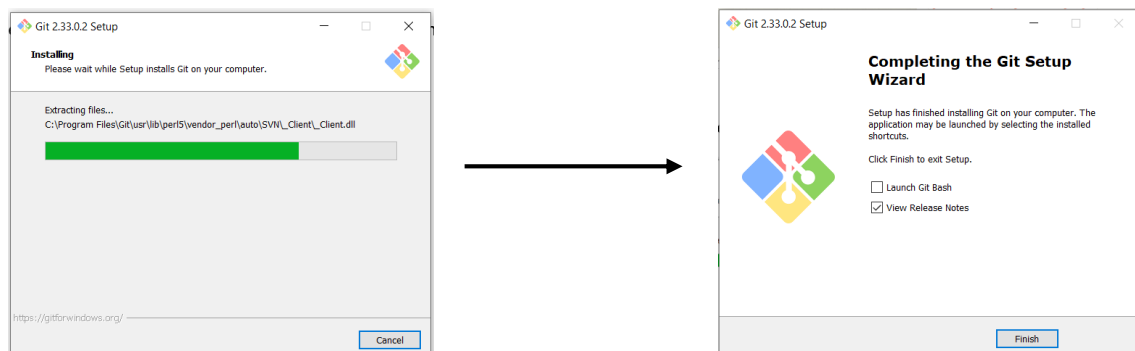
En este caso, lo descargaremos para Windows.



2.2 Instalación

Una vez descargado, aceptamos que el programa haga cambios en el dispositivo y pasamos "Next" todo por defecto.

Se completa rápidamente y finalizamos la instalación.

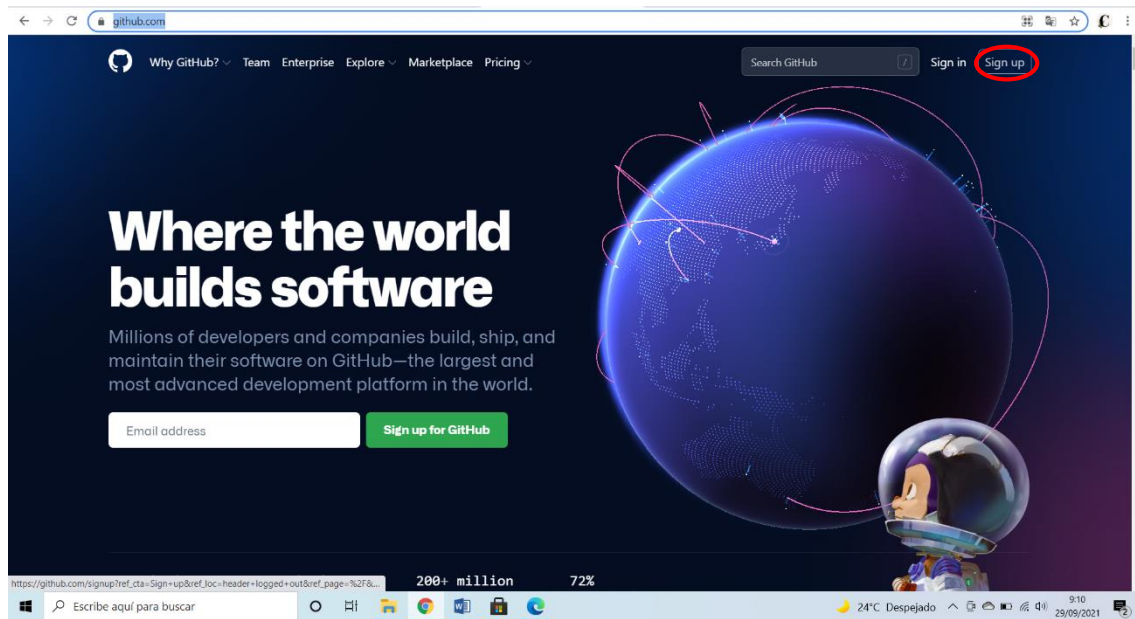


2.3 Creación y configuración de la cuenta

Para comenzar a crear el repositorio, primero tenemos que crearnos una cuenta GitHub.

Para ello, entraremos en la página oficial de GitHub: <https://github.com/>

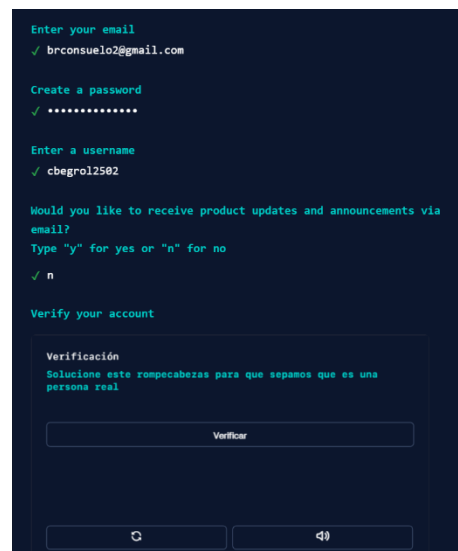
Seguidamente, en la parte superior derecha de la página, entramos en “Sing Up”



Nos pedirán una cuenta de gmail, la creación de una contraseña, el nombre que queremos poner de usuario y si queremos que nos envíen información o anuncios.

Solucionamos un rompecabezas y así, verificamos la cuenta.

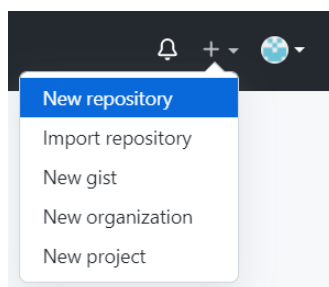
Ahora, ya tenemos cuenta de GitHub.



3 Creación de un nuevo repositorio

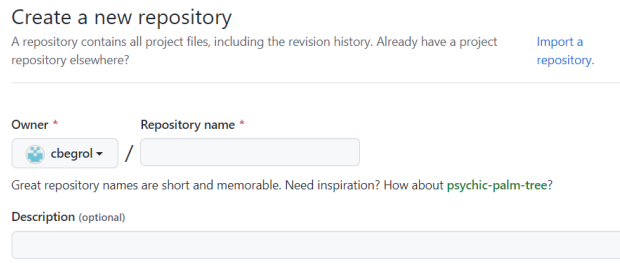
3.1 Paso 1:

En la esquina superior derecha de la página, utiliza el menú desplegable y selecciona “Repositorio Nuevo”.



3.2 Paso 2:


En el menú desplegable de “Propietario”, selecciona la cuenta en la cual quieres crear el repositorio. Además, hay que escribir un nombre para el proyecto y te da la opción de escribir una descripción.



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 *

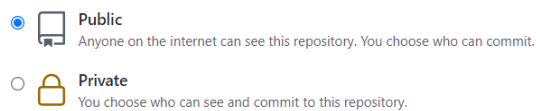
 cbegrol /


Great repository names are short and memorable. Need inspiration? How about [psychic-palm-tree?](#)


Description (optional)

3.3 Paso 3:

Elige la visibilidad del repositorio.

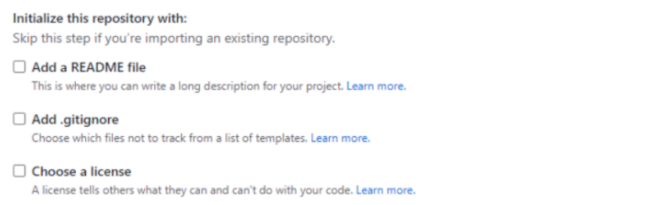


☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

3.4 Paso 4:

Además, nos da la opción de elegir si queremos crear el repositorio añadiendo un Readme.md, .gitignore o licencia.



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

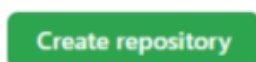
☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **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.](#)





3.5 Paso 5:

Da clic en “Crear repositorio”.



Cuando ya hemos creado el repositorio, lo primero que nos sale es la lista de comandos git con los que vamos a trabajar a continuación.


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

 Set up in Desktop or  HTTPS  SSH <https://github.com/cbegrol/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/cbegrol/proyecto01.git
git push -u origin main
```




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

```
git remote add origin https://github.com/cbegrol/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.



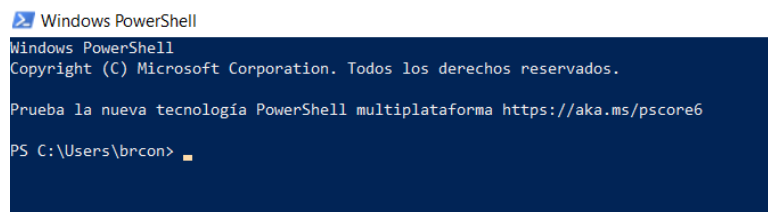
4 Windows PowerShell

Una vez aparezcan los comandos git, podemos pasar a otro programa llamado Windows PowerShell.

4.1 Abrir programa

Abrimos PowerShell para poder crear el repositorio y comenzar con la secuencia de comandos.

Una vez lo abrimos, nos aparece esto:



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

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

PS C:\Users\brcon>
```

4.2 Comando cd

Comenzamos con el comando cd, uno de los comandos más esenciales de la consola de Windows. Sirve para cambiar de directorio, utilizando la fórmula `*cd < RutaDirectorio >*` para ir al directorio o carpeta concreta que le digas, o `*cd..*` (con dos puntos) para salir de una carpeta e ir al nivel superior o carpeta donde estaba alojada.

Así que el primer comando a utilizar es este ya que hay que ubicarse en el directorio donde tenemos los archivos que vamos a subir al directorio.


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

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01>
```

4.3 Comando git init

Git init es un comando que se utiliza una sola vez durante la configuración inicial de un repositorio nuevo. Al ejecutar este comando, se creará un nuevo subdirectorio .git en tu directorio de trabajo actual. También se creará una nueva rama principal.

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

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git init
Reinitialized existing Git repository in C:/Users/brcon/Documents/1ºDAM/Proyecto01/.git/
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01>
```

4.4 Comando git status

El comando git status muestra el estado del directorio de trabajo y del área del entorno de ensayo. Permite ver los cambios que se han preparado, los que no y los archivos en los que Git no va a realizar el seguimiento. El resultado del estado no muestra ninguna información relativa al historial del proyecto.

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

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git init
Reinitialized existing Git repository in C:/Users/brcon/Documents/1ºDAM/Proyecto01/.git/
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01>
```

4.5 Comando git add .

Git add . es una pieza fundamental del flujo básico de git ya que es el comando que mueve al índice las modificaciones que hayamos realizado. El índice es un snapshot del contenido del área de trabajo en un momento dado.

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

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git init
Reinitialized existing Git repository in C:/Users/brcon/Documents/1ºDAM/Proyecto01/.git/
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git add .
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01>
```

4.6 Comando git commit -m

El comando git commit guardará todos los cambio hechos en la zona de montaje o área de preparación (staging area), junto con una breve descripción del usuario, en un "commit" al repositorio local.

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

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git init
Initialized empty Git repository in C:/Users/brcon/Documents/1ºDAM/Proyecto01/.git/
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git add .
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git commit -m "primer commit"
[master (root-commit) 19f0d16] primer commit
4 files changed, 43 insertions(+)
create mode 100644 README.md.txt
create mode 100644 doc/REPOSITORIO GITHUB.docx
create mode 100644 doc/~$POSITORIO GITHUB.docx
create mode 100644 src/programa.txt.txt
```

4.7 Comando git config --global user.email y comando git config --global user.name

El comando git config es una función útil que sirve para definir valores de configuración de Git a nivel de un proyecto global o local.

En el comando git config --global user.email, introduciremos nuestra dirección de correo.

En el comando git config --global user.name, introduciremos nuestro nombre.

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

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git init
Initialized empty Git repository in C:/Users/brcon/Documents/1ºDAM/Proyecto01/.git/
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git add .
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git commit -m "primer commit"
[master (root-commit) 19f0d16] primer commit
 4 files changed, 43 insertions(+)
 create mode 100644 README.md.txt
 create mode 100644 doc/REPOSITORIO GITHUB.docx
 create mode 100644 doc/~$POSITARIO GITHUB.docx
 create mode 100644 src/programa.txt.txt
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git config --global user.email "cbegrol2502@g.educaand.es"
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git config --global user.name "cbegrol"
```

4.8 Comando git branch -M main

El comando git branch te permite crear, enumerar y eliminar ramas, así como cambiar su nombre. No te permite cambiar entre ramas o volver a unir un historial bifurcado.

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

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git init
Initialized empty Git repository in C:/Users/brcon/Documents/1ºDAM/Proyecto01/.git/
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git add .
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git commit -m "primer commit"
[master (root-commit) 19f0d16] primer commit
 4 files changed, 43 insertions(+)
 create mode 100644 README.md.txt
 create mode 100644 doc/REPOSITORIO GITHUB.docx
 create mode 100644 doc/~$POSITARIO GITHUB.docx
 create mode 100644 src/programa.txt.txt
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git config --global user.email "cbegrol2502@g.educaand.es"
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git config --global user.name "cbegrol"
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git branch -M main
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01>
```

4.9 Comando git remote add origin

Este es un comando que dice "Empuje las confirmaciones en la rama local llamada master al remoto llamado "Origin ". Una vez que esto se haya ejecutado, todas las cosas que sincronizó por última vez con Origin se enviarán al repositorio remoto y otras personas podrán verlas allí.

```
Windows PowerShell
copyright (C) Microsoft Corporation. Todos los derechos reservados.

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git init
Initialized empty Git repository in C:/Users/brcon/Documents/1ºDAM/Proyecto01/.git/
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git add .
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git commit -m "primer commit"
[master (root-commit) 19f0d16] primer commit
 4 files changed, 43 insertions(+)
 create mode 100644 README.md.txt
 create mode 100644 doc/REPOSITORIO GITHUB.docx
 create mode 100644 doc/~$POSITORIO GITHUB.docx
 create mode 100644 src/programa.txt.txt
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git config --global user.email "cbegrol2502@g.educaand.es"
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git config --global user.name "cbegrol"
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git branch -M main
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git remote add origin https://github.com/cbegrol/proyecto01..git
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01>
```

4.10 Comando git push -u origin main

Este comando te permite subir los commits desde tu rama (branch) local en tu repositorio git local al repositorio remoto. Para poder subir a tu repositorio remoto, te debes asegurar de hacer commit a todos tus cambios al repositorio local.

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

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

PS C:\Users\brcon> cd C:\Users\brcon\Documents\1ºDAM\Proyecto01
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git init
Initialized empty Git repository in C:/Users/brcon/Documents/1ºDAM/Proyecto01/.git/
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git status
On branch master

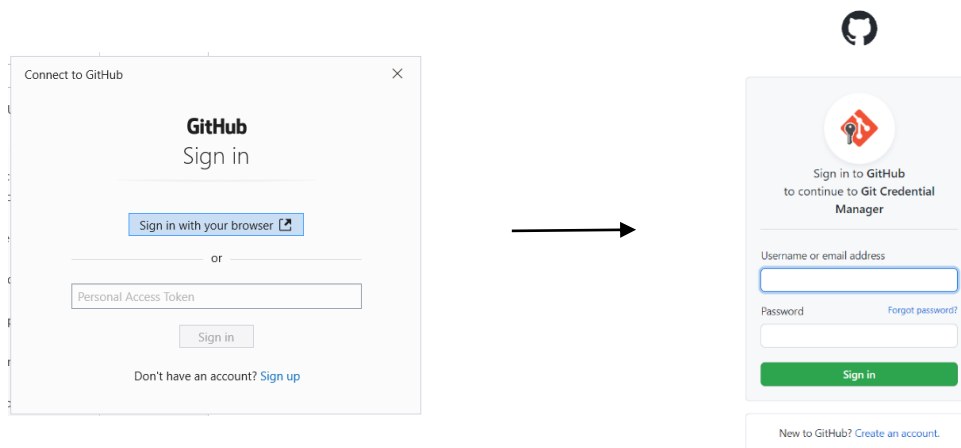
No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git add .
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git commit -m "primer commit"
[master (root-commit) 19f0d16] primer commit
4 files changed, 43 insertions(+)
 create mode 100644 README.md.txt
 create mode 100644 doc/REPOSITORIO GITHUB.docx
 create mode 100644 doc/~$POSITARIO GITHUB.docx
 create mode 100644 src/programa.txt.txt
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git config --global user.email "cbegrol2502@g.educaand.es"
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git config --global user.name "cbegrol"
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git branch -M main
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git remote add origin https://github.com/cbegrol/proyecto01.git
PS C:\Users\brcon\Documents\1ºDAM\Proyecto01> git push -u origin main
```

5 Conexión de cuentas y finalización del proyecto

Al hacer push -u origin main, solicita la conexión con GitHub y autentica tu cuenta



Finalmente, al autorizar el Git Credential Manager, todos los archivos que tenemos en nuestra carpeta y la información que queremos subir, se completará automáticamente.

Si en algún momento se realizan cambios en la carpetas o en los documentos, se puede actualizar la infoamción en el repositorio, añadiendo a la lista los comando git add ., git commit -m "nuevo nombre" y git push -u origin main.