

## Actividad 1.

¿Cuáles son los inconvenientes y beneficios de trabajar con cuentas de GitHub gratuitas?

Trabajar con cuentas GitHub gratuitas ofrece una serie de inconvenientes y beneficios:

### Inconvenientes:

- Limitaciones en repositorios privados: Las cuentas gratuitas tienen restricciones en la cantidad de repositorios privados que puedes crear.
- Soporte técnico limitado: Con una cuenta gratuita, es posible que no recibas el mismo nivel de soporte técnico que con una cuenta de pago.
- Funcionalidades avanzadas restringidas: Algunas características avanzadas, como la integración con herramientas de gestión de proyectos o la capacidad de establecer políticas de acceso más granulares, pueden no estar disponibles en cuentas gratuitas.

### Beneficios:

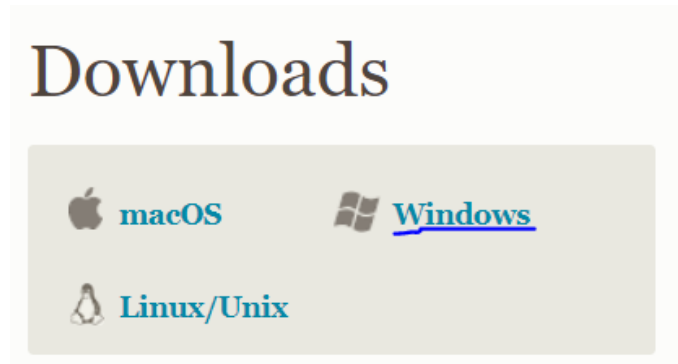
- Costo: La principal ventaja de utilizar una cuenta gratuita es que no tienes que pagar nada por acceder a las funciones básicas de GitHub.
- Colaboración: Aunque las cuentas gratuitas tienen algunas limitaciones en cuanto a la colaboración en proyectos privados, aún puedes trabajar con otros colaboradores en proyectos públicos de forma gratuita.
- Control de versiones: GitHub proporciona una sólida plataforma para controlar versiones de tu código, lo que facilita el seguimiento de cambios, la resolución de conflictos y la colaboración con otros desarrolladores, todo lo cual está disponible en cuentas gratuitas.

## Actividad 2.

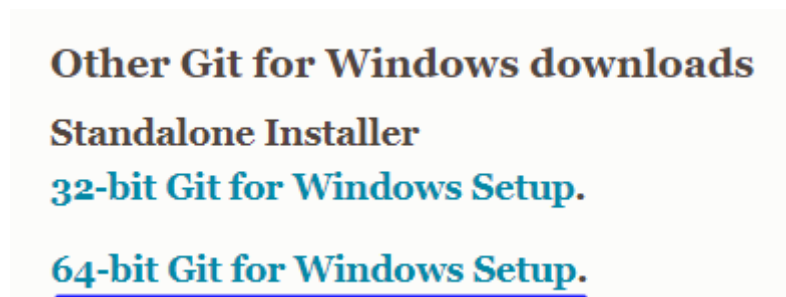
Realiza un manual explicativo en el que se demuestre cómo instalar y configurar Git en un equipo cliente. Crear un repositorio dentro de Github. Subir directamente en Github algún código que tengáis hecho y os lo bajéis a vuestro equipo.

Lo primero que hay que hacer es ir a la página de Git <https://git-scm.com/downloads>

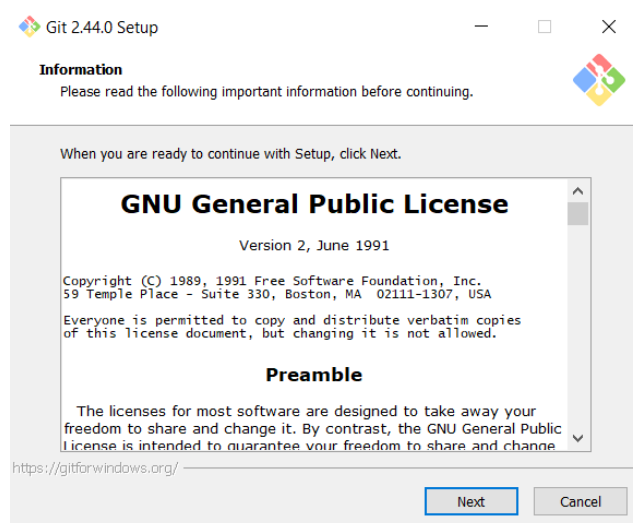
Luego le damos a Downloads y a Windows

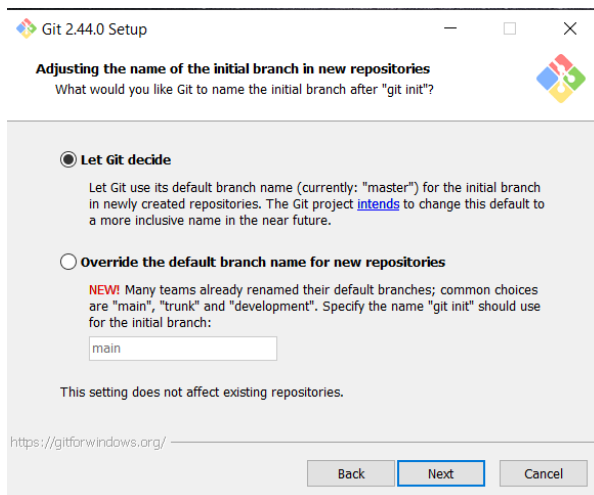
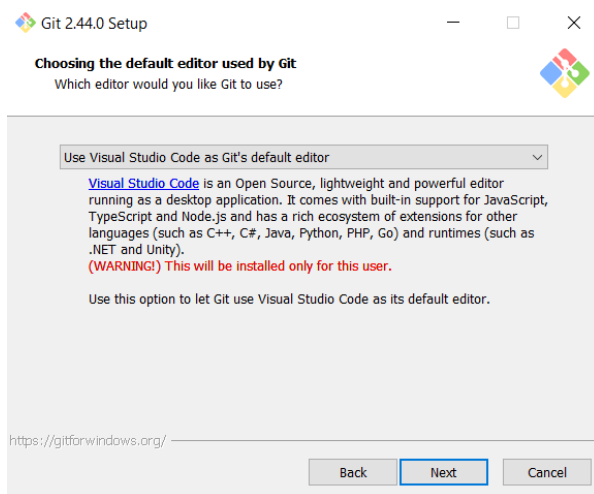
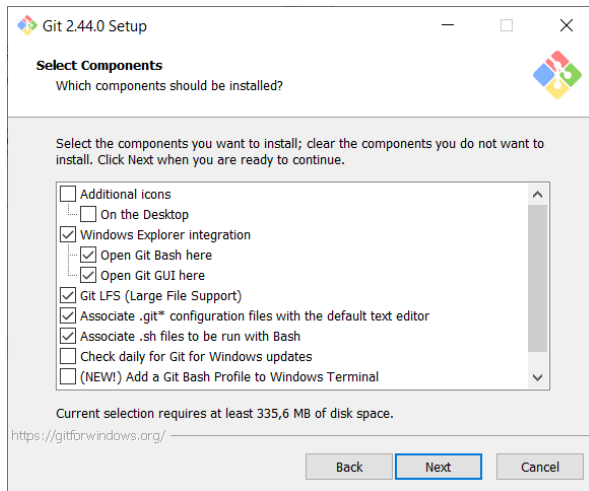


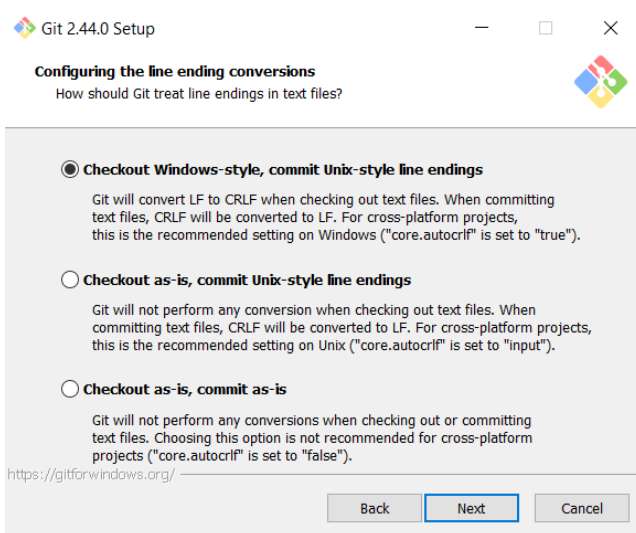
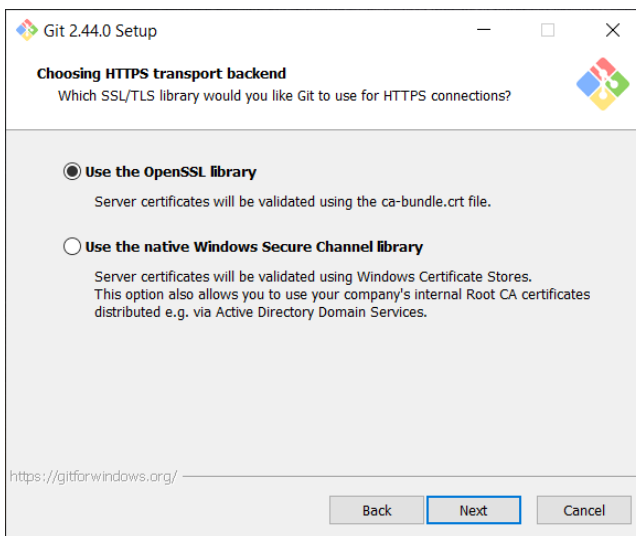
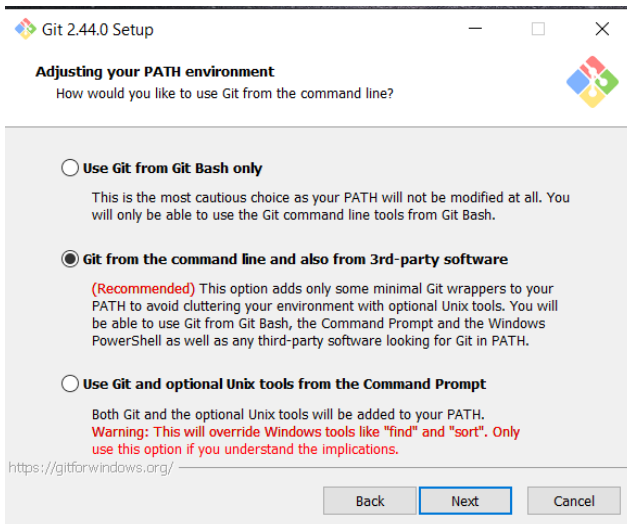
Elegiremos la opción de 64 bits

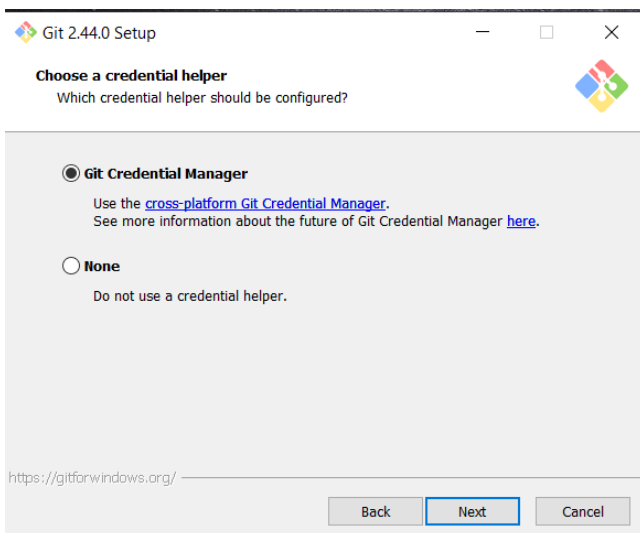
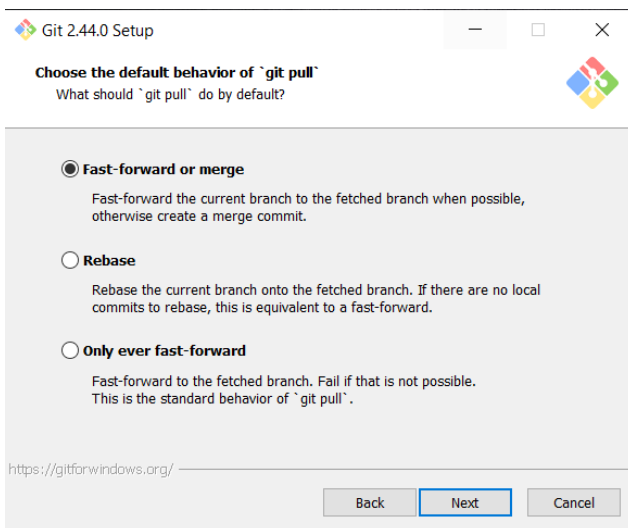
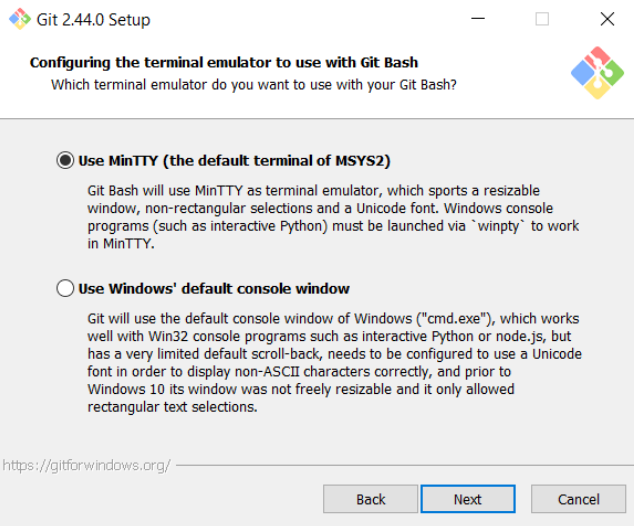


Cuando se descargue ejecutaremos la instalación y vamos a marcar lo que salga en las capturas



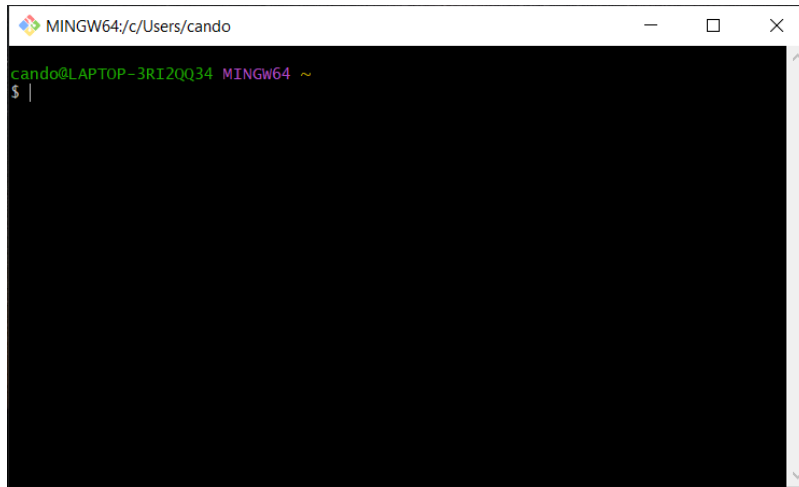






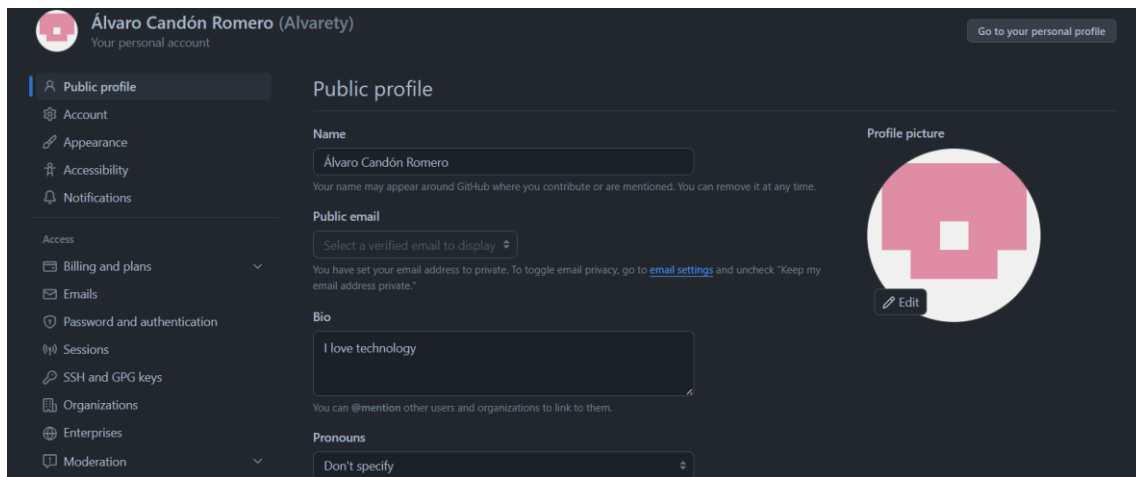
Una vez terminado de escoger opciones le daremos a Instalar

Una vez finalizado marcaremos abrirlo en Git Bash y nos saldra esto

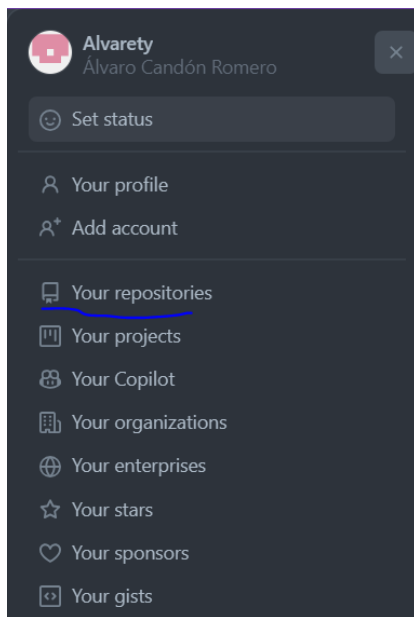


```
MINGW64; c:/Users/cando
cando@LAPTOP-3RI2QQ34 MINGW64 ~
$ |
```

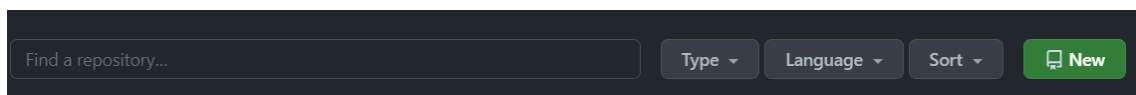
Para crear un repositorio en GitHub hay que crearse una cuenta, la cual me acabo de hacer



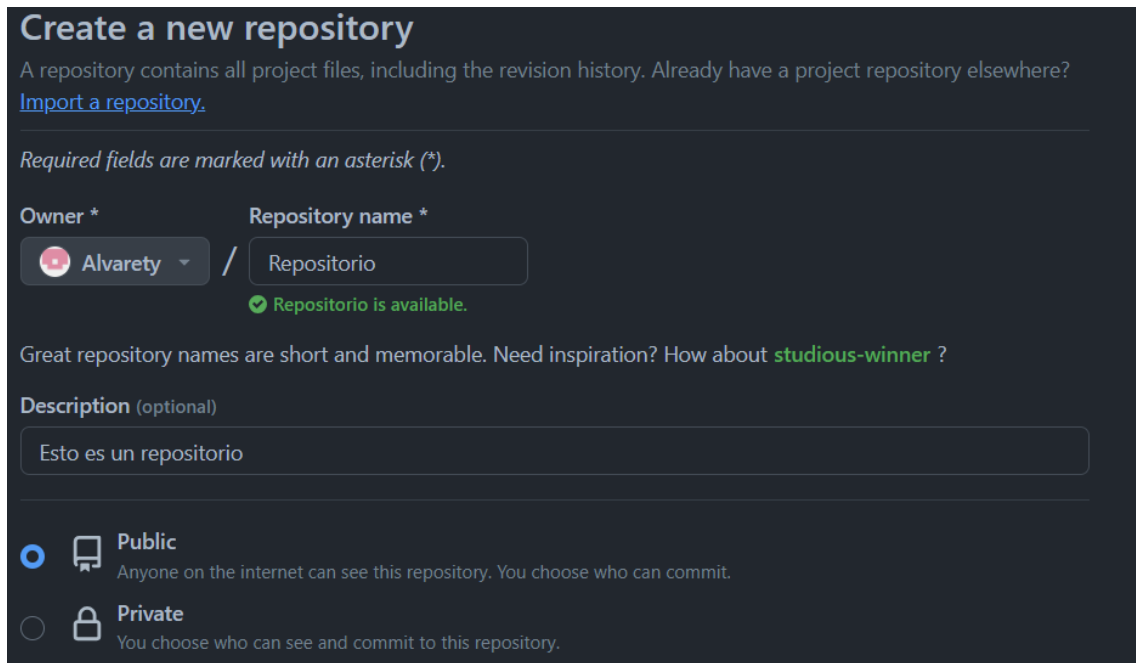
Nos vamos al menú y nos vamos a tus repositorios.



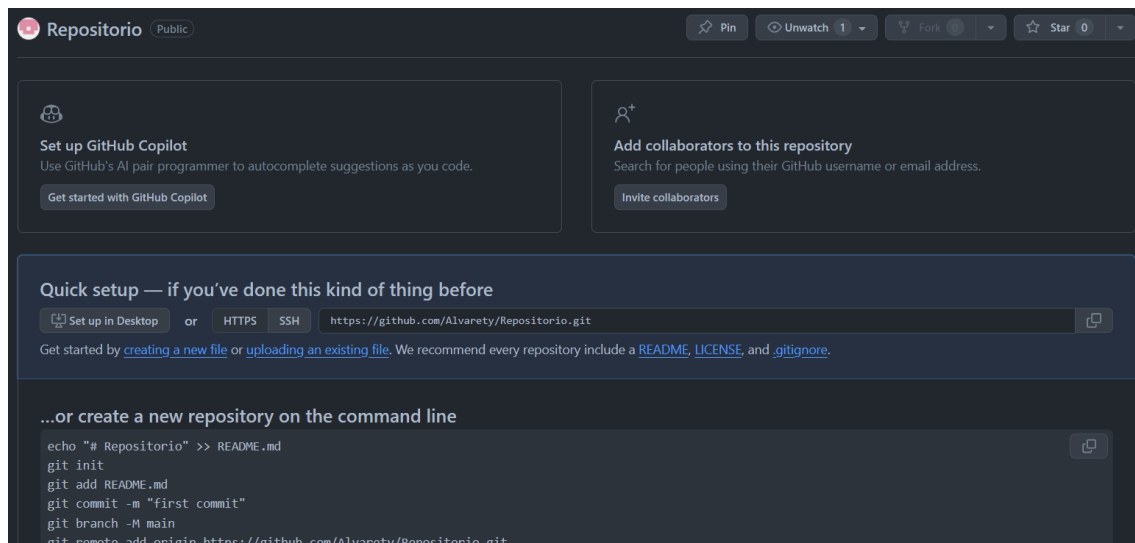
Crearemos un nuevo repositorio dándole a New



Le damos un nombre y lo dejamos en público



Creamos el repositorio y aquí lo veremos ya creado con sus funciones



Para subir un proyecto a un repositorio primero debemos hacer un GitBash en la carpeta donde tenemos el proyecto.

Ahora vamos a iniciar el GIT, una vez iniciado pondremos git status para ver el estado del proyecto.

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios
$ git init
Initialized empty Git repository in C:/Users/cando/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios/.git/

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Ejercicio1_ACR.java

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

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ |
```

Ahora añadiremos cositas como el correo o el nombre, lo cual veremos en estas capturas.

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ git add .

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ git commit -m "Se agrega el Proyecto"
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 'cando@LAPTOP-3RI2QQ34.(none)')

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ |
```



```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ git config --global user.email candonromeroalvaro2@gmail.com

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ git config --global user.name Alvarety
```

Repito el comando de “Se agrega el Proyecto” y ahí ya estará agregado

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ git commit -m "Se agrega el Proyecto"
[master (root-commit) 9ae84cf] Se agrega el Proyecto
1 file changed, 20 insertions(+)
create mode 100644 Ejercicio1_ACR.java
```

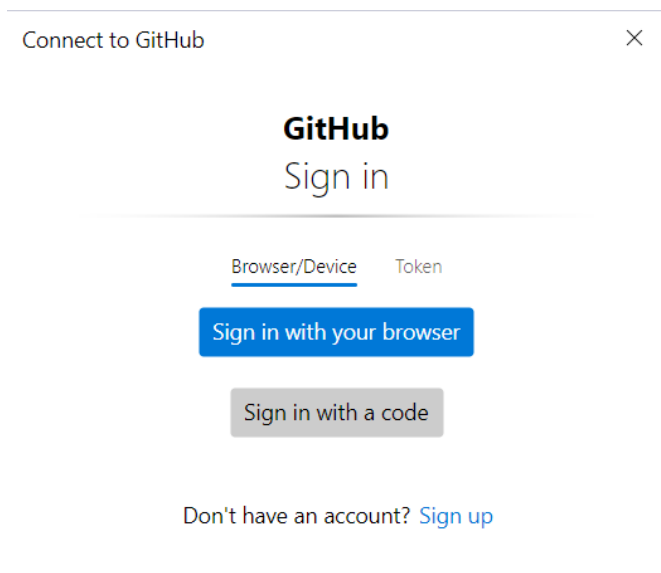
Añado el proyecto al repositorio

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (master)
$ git branch -M main

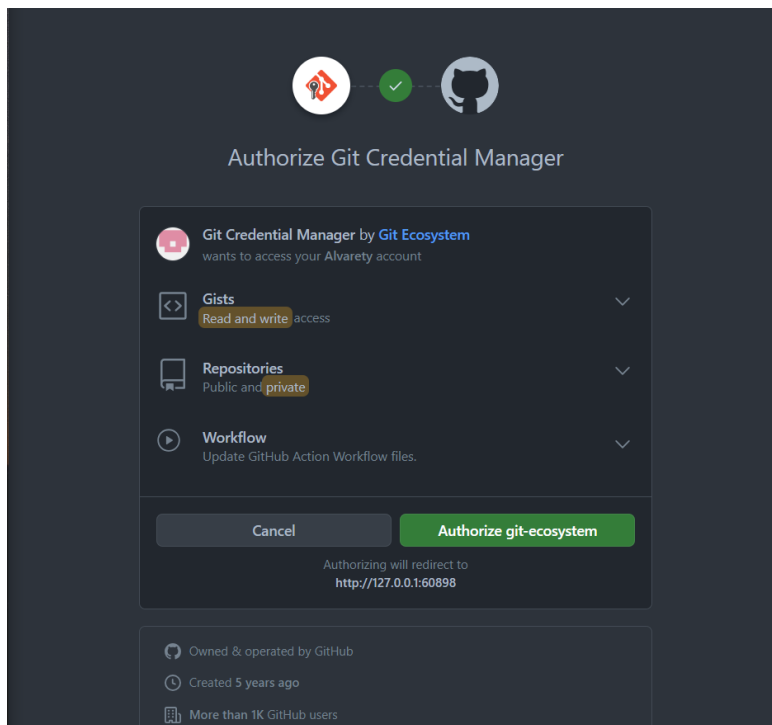
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (main)
$ git remote add origin https://github.com/Alvarety/Repositorio.git

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programación/Array/Ejercicios (main)
$ git push -u origin main
```

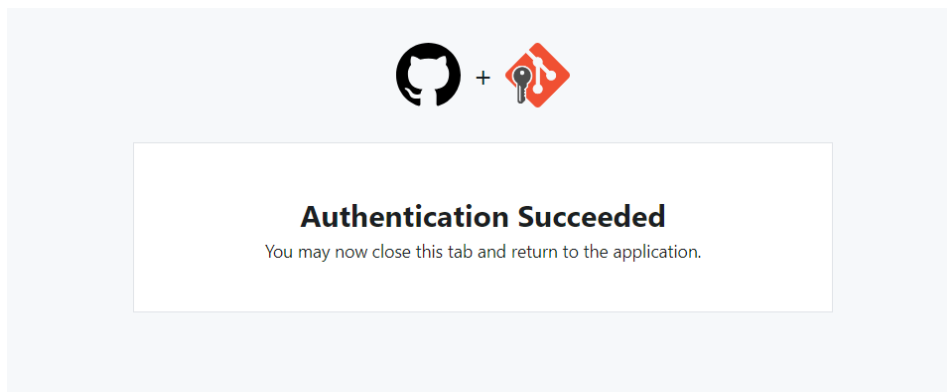
Ahora nos sale esta ventana donde podemos conectarnos con GitHub



Vamos a autorizar la sincronización



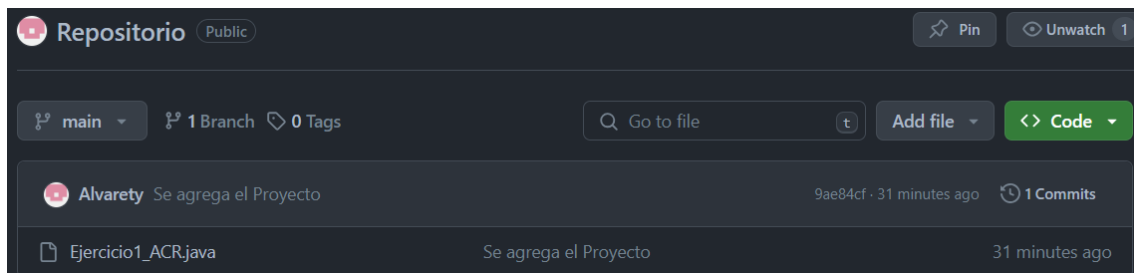
Ponemos contraseña de GitHub, y ya la tendremos autorizada



Con esto se puede ver que se autorizó la subida del proyecto.

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/1º DAW/Programacion/Array/Ejercicios (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 512 bytes | 512.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Alvarety/Repositorio.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Aquí tenemos el proyecto subido.



### Actividad 3.

Trabajando en pareja debéis crear un proyecto en Github con varias clases/ficheros. Ambos debéis crearle una rama y descargarla en vuestros equipos. Hacer modificaciones en ficheros diferentes. Sincronizar esos cambios en el repositorio central de Github y que el otro miembro del equipo se descargue dichos cambios.

Como no tengo pareja para hacer este ejercicio, lo hare solo.

Con nuestra cuenta de GitHub, creamos un nuevo repositorio pero esta vez será para hacer este ejercicio.

A screenshot of the 'Create a new repository' form on GitHub. The title 'Create a new repository' is at the top. Below it, a subtitle says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. A note states 'Required fields are marked with an asterisk (\*)'. The form has two main sections: 'Owner \*' with a dropdown menu showing 'Alvarety' and 'Repository name \*' with a text input field containing 'Repositorio Doble'. A green checkmark message says 'Your new repository will be created as Repositorio-Doble. The repository name can only contain ASCII letters, digits, and the characters ., -, and \_.' Below this, a suggestion says 'Great repository names are short and memorable. Need inspiration? How about [miniature-octo-spoon](#) ?'. There is a 'Description (optional)' text area. At the bottom, there are two radio button options: 'Public' (selected) with the description 'Anyone on the internet can see this repository. You choose who can commit.' and 'Private' with the description 'You choose who can see and commit to this repository.'

Ahora clonaremos el repositorio en el ordenador, así crearemos una copia local del repositorio del ordenador.

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio
$ git clone https://github.com/Alvarety/Repositorio-Doble
```

Cambiaremos al directorio del repositorio clonado y crearemos una nueva rama

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio
$ cd Repositorio-Doble

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (main)
$ git checkou -b mi_rama
git: 'checkou' is not a git command. See 'git --help'.

The most similar command is
    checkout

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (main)
$ git checkout -b mi_rama
Switched to a new branch 'mi_rama'
```

Ahora crearemos varios archivos

 Uno		09/03/2024 12:53	Archivo JAVA
 Dos		09/03/2024 12:56	Archivo JAVA

Ahora usaremos git add . para agregar todos los cambios y luego git commit -m "Mensaje del commit" para hacer un commit de los cambios en tu rama.

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (mi_rama)
$ git add .

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (mi_rama)
$ git commit -m "Agrego archivos y modificaciones"
[mi_rama (root-commit) d599bc3] Agrego archivos y modificaciones
2 files changed, 41 insertions(+)
create mode 100644 Dos.java
create mode 100644 Uno.java
```

Ahora subiremos mi rama al repositorio remoto

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (mi_rama)
$ git push origin mi_rama
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 831 bytes | 831.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Alvarety/Repositorio-Doble
* [new branch]      mi_rama -> mi_rama
```

Ahora crearemos una nueva rama simulando que es la de mi compañero

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (mi_rama)
$ git checkout -b rama_compa
Switched to a new branch 'rama_compa'
```



09/03/2024 13:09

Archivo JAVA

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (rama_compa)
$ git add .

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (rama_compa)
$ git commit -m "Agrego archivos y modificaciones"
[rama_compa b955e59] Agrego archivos y modificaciones
2 files changed, 11 insertions(+), 3 deletions(-)
create mode 100644 Tres.java
```

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (rama_compa)
$ git push origin rama_compa
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 532 bytes | 532.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'rama_compa' on GitHub by visiting:
remote:   https://github.com/Alvarety/Repositorio-Doble/pull/new/rama_compa
remote:
To https://github.com/Alvarety/Repositorio-Doble
 * [new branch]      rama_compa -> rama_compa
```

Ahora sincronizaremos los cambios del compañero.

```
cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (rama_compa)
$ git checkout mi_rama
Switched to branch 'mi_rama'

cando@LAPTOP-3RI2QQ34 MINGW64 ~/OneDrive/Escritorio/Repositorio-Doble (mi_rama)
$ git pull origin rama_compa
From https://github.com/Alvarety/Repositorio-Doble
 * branch      rama_compa -> FETCH_HEAD
Updating d599bc3..b955e59
Fast-forward
 Dos.java    | 3 ---
 Tres.java   | 11 ++++++++
 2 files changed, 11 insertions(+), 3 deletions(-)
 create mode 100644 Tres.java
```