



**CICLO: DAW
MÓDULO DE DAW**

Tarea N°6

**Alumno:
Borja Arques Amat
20908598B**

Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.

Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.

Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenece a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.

Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.

Contenido

1. Documentos que se adjuntan a este informe.2
2. Resto de epígrafes que componen el desarrollo de este informe.2

1. Documentos que se adjuntan a este informe.

A continuación, se detallan los documentos que componen la presente entrega de la tarea:

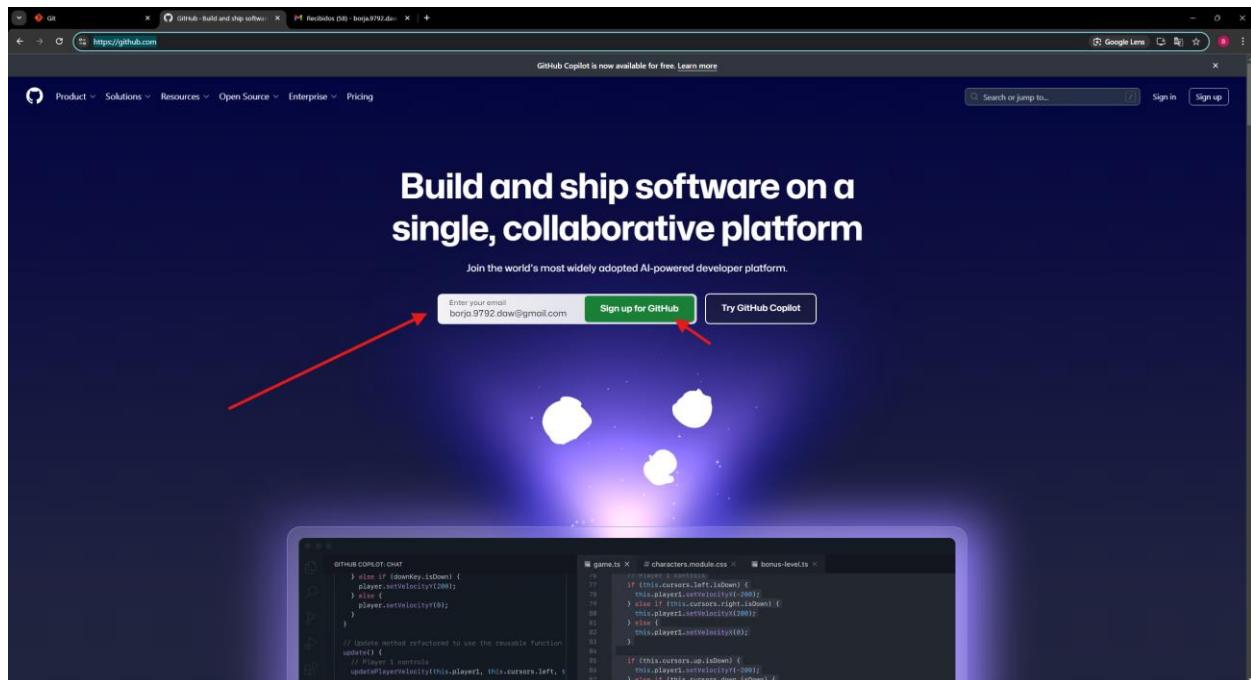
1. Informe de elaboración de la tarea.

2. Resto de epígrafes que componen el desarrollo de este informe.
3. (RA6_d) Se han utilizado herramientas colaborativas para la elaboración y mantenimiento de la documentación.

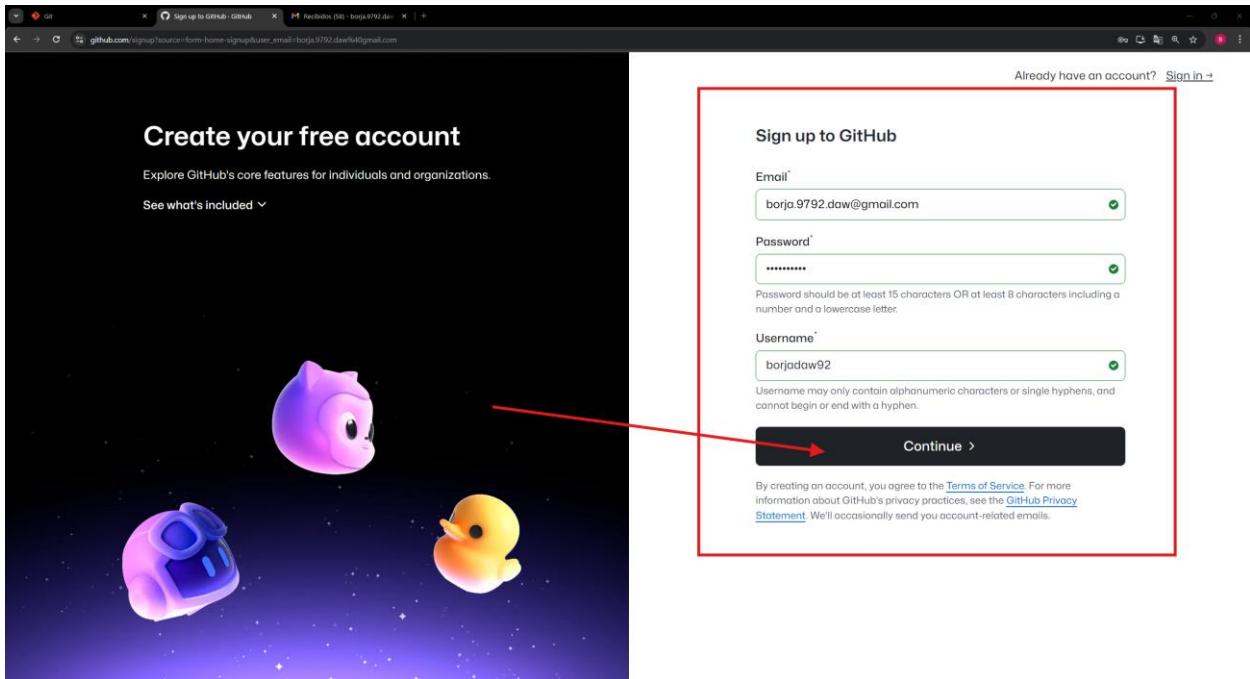
Para hacer uso del repositorio remoto que nos ofrece GitHub, lo primero que deberíamos de hacer seria registrarnos en la página:

<https://github.com>

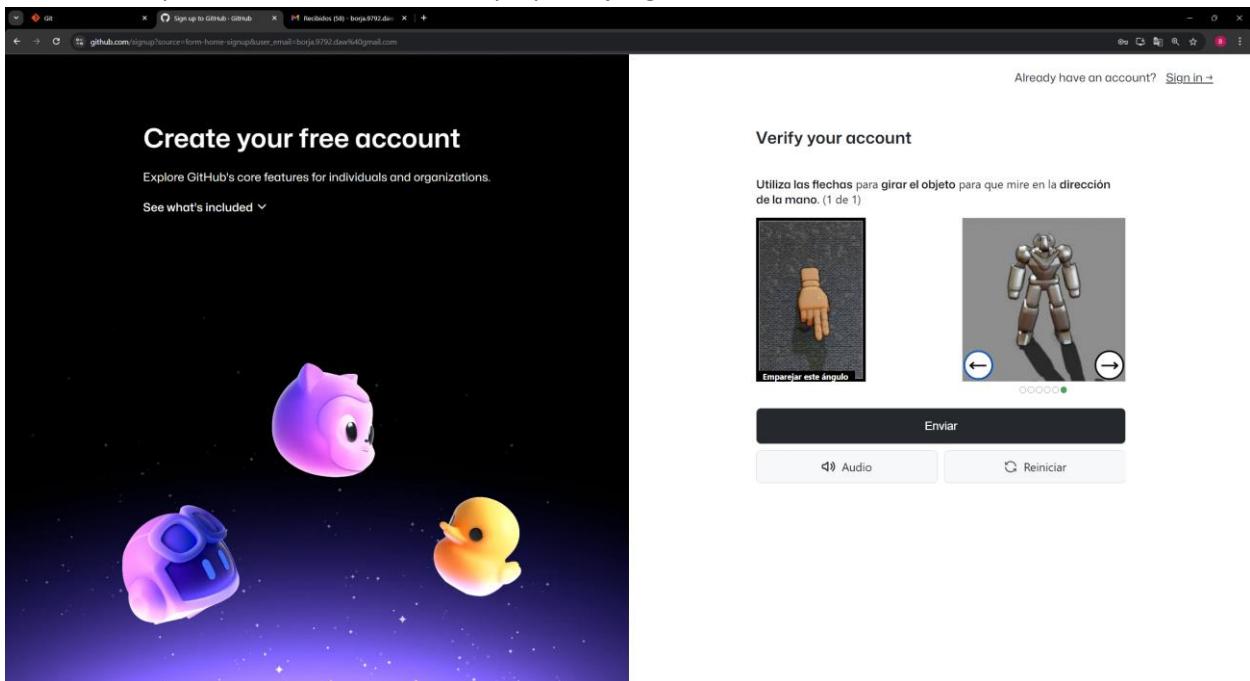
introducimos el correo con el que nos vamos a registrar y clic en Sign up for GitHub:



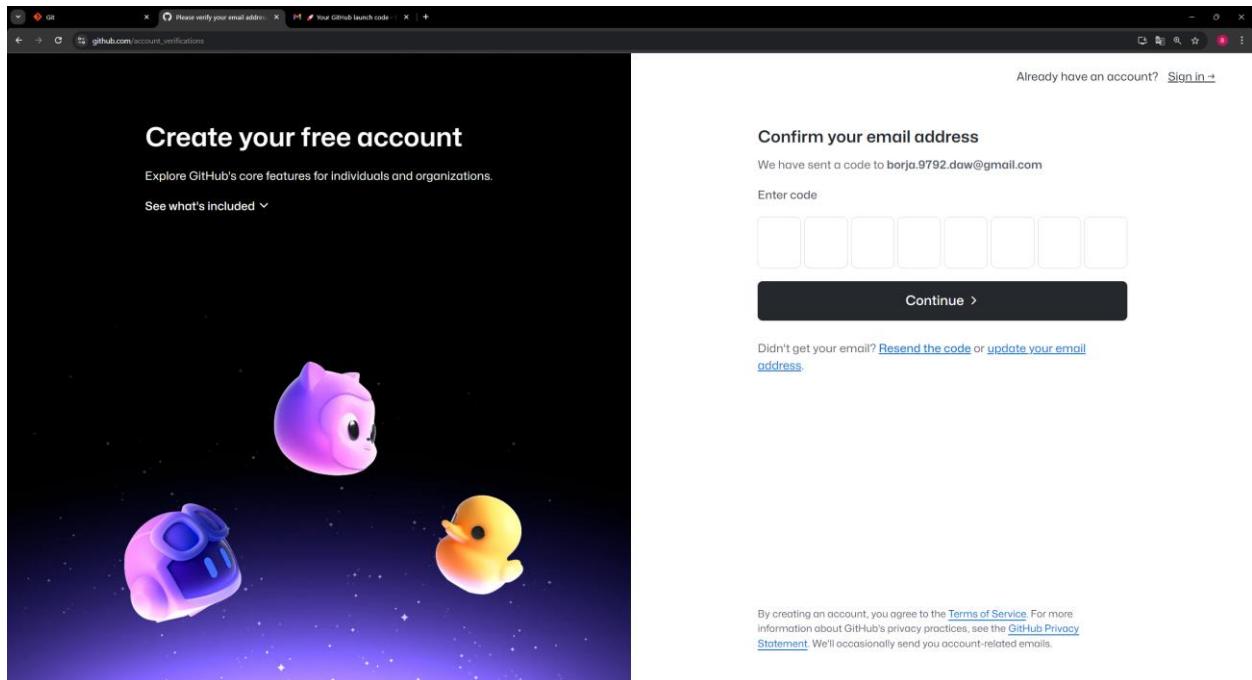
Rellenamos los campos y clic en Continue>



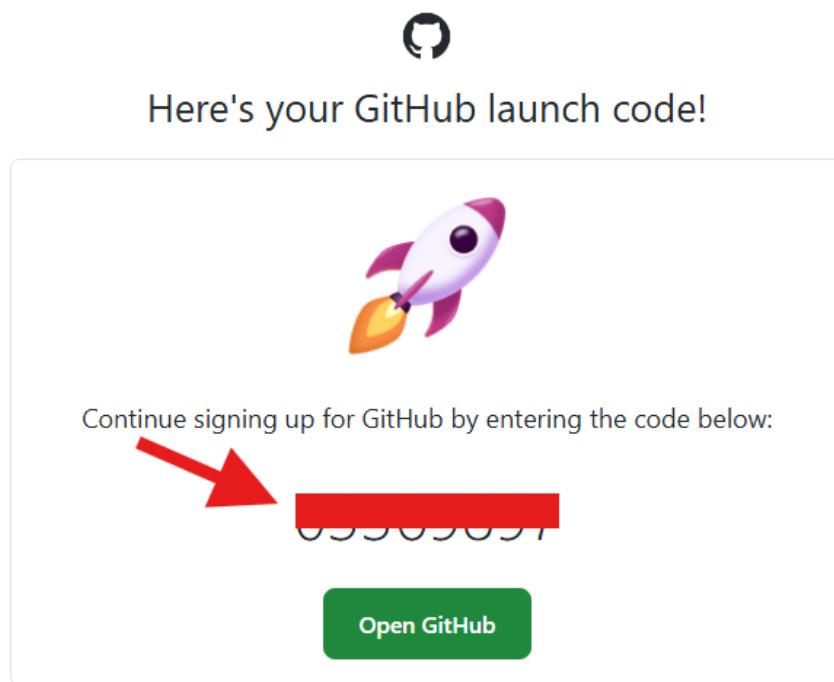
Verificamos que somos humanos con un pequeño juego:



nos mandaran un código a nuestro correo de registro:

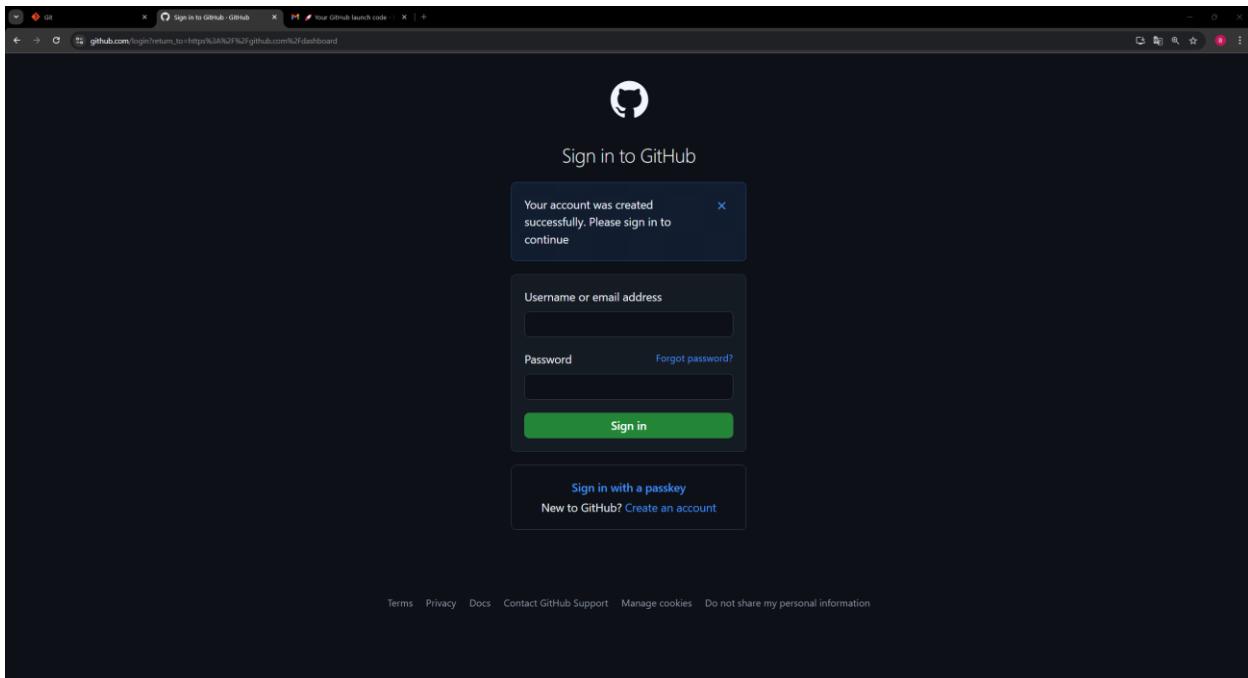


Por seguridad voy a taparlo, pero en nuestro email nos habrán mandado un email como este. Este será el código:

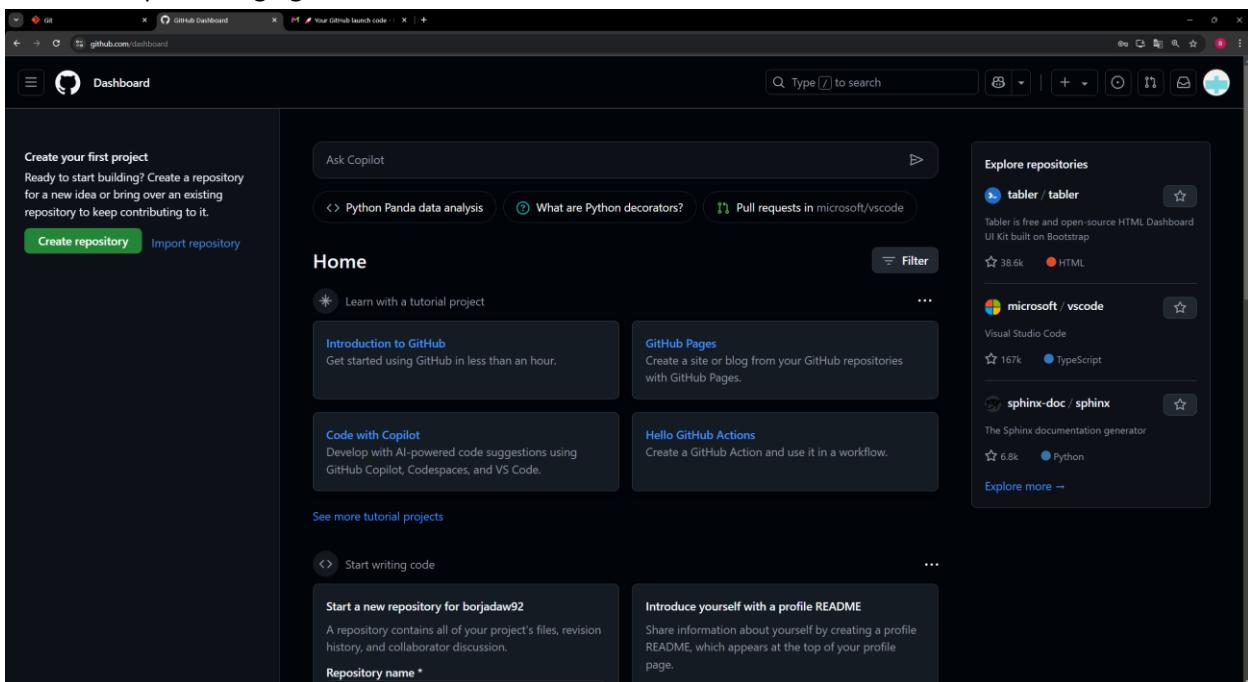


Not able to enter the code? Paste the following link into your browser:
https://github.com/account_verifications/confirm/1c84f5ca-b11f-408c-82fb-3fdcea b2689d/03369897

Al introducirlo, nos manda un mensaje de que nos hemos registrado correctamente y ya podríamos logear por primera vez:



La vista del primer logging:



Para crear nuestro primer repositorio remoto hacemos clic en “Create repository”:

Introducimos el nombre del repositorio que queremos crear, yo para la tarea le he nombrado Tarea6DAW y lo pongo publico ya que necesito que lo pueda revisar el profesor:

Create a new repository

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

Required fields are marked with an asterisk (*).

Owner * Repository name * Tarea6DAW is available.

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

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:

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

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Clic en Create repository para crearlo definitivamente:

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:

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

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

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

① You are creating a public repository in your personal account.

© 2025 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

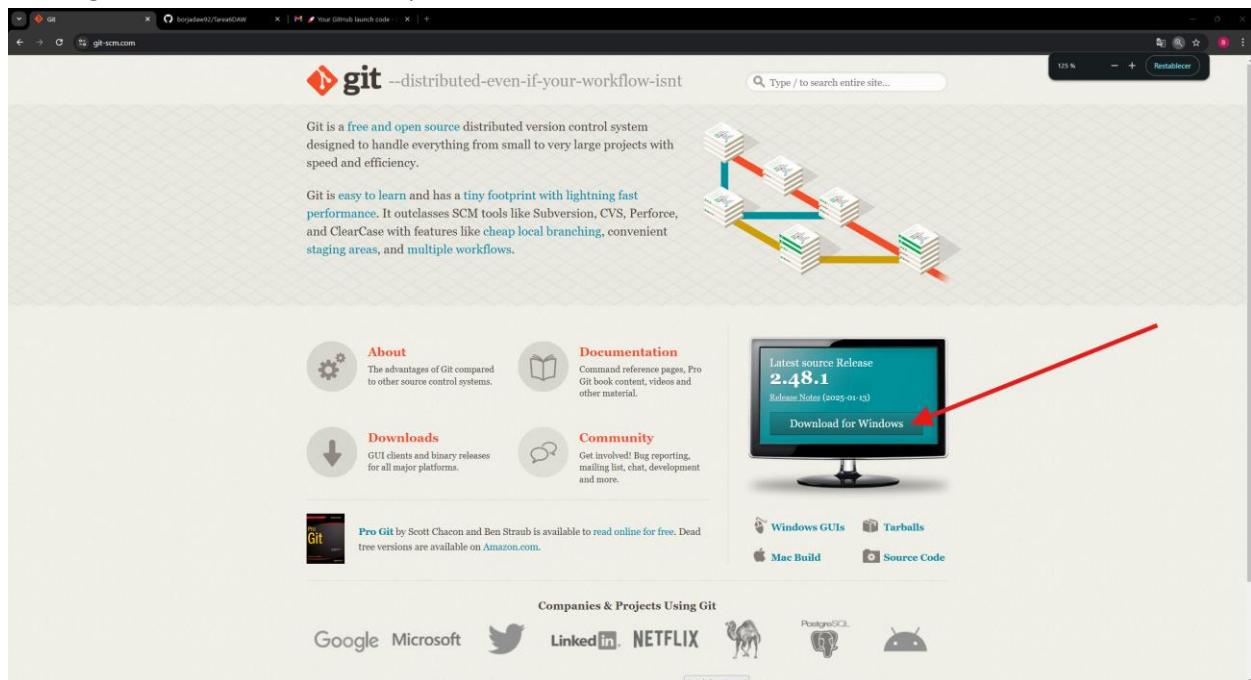
4. (RA6_e) Se ha instalado, configurado y utilizado un sistema de control de versiones.

En este apartado vamos a proseguir con la instalación de git en nuestra máquina:

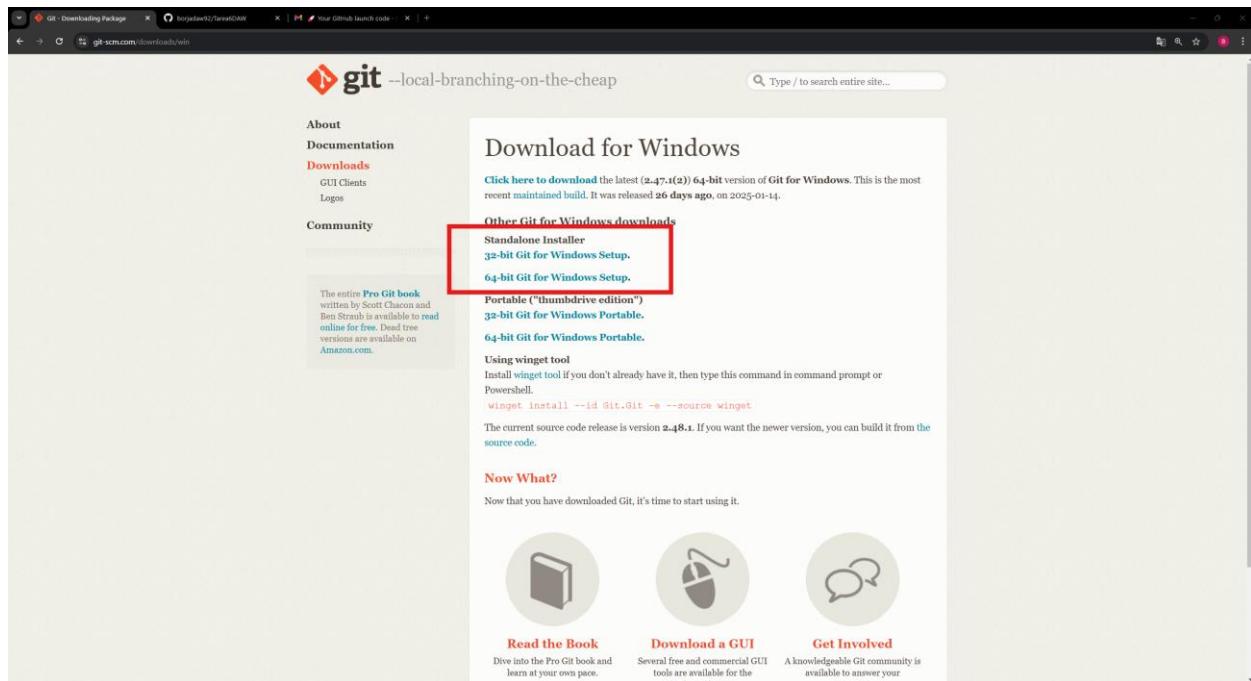
Nos dirigimos al siguiente enlace:

<https://git-scm.com>

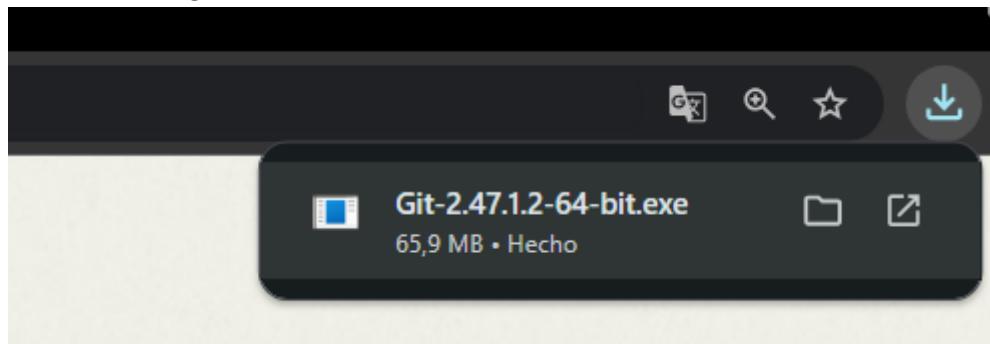
En mi caso como me lo estoy instalando y configurando para mi propio equipo y usarlo en un futuro, me descargo la última versión 2.48.1 para Windows:



Seleccionamos o la versión de 64 o la de 32 según nuestra máquina:



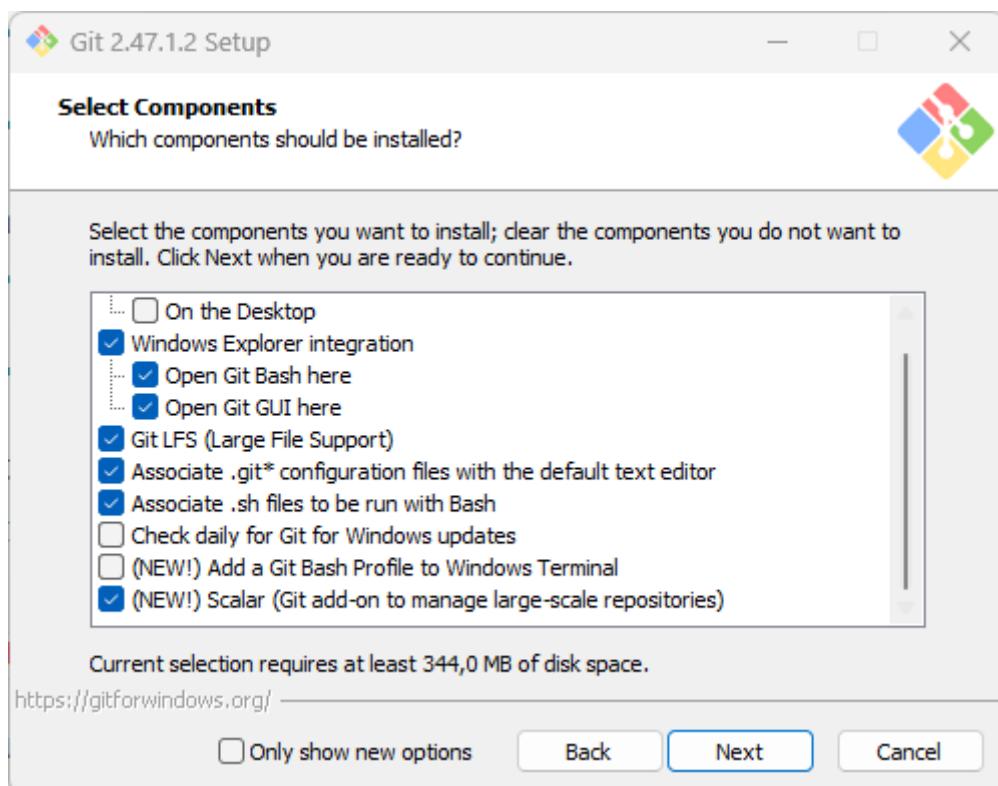
Una vez descargado, lo lanzamos:



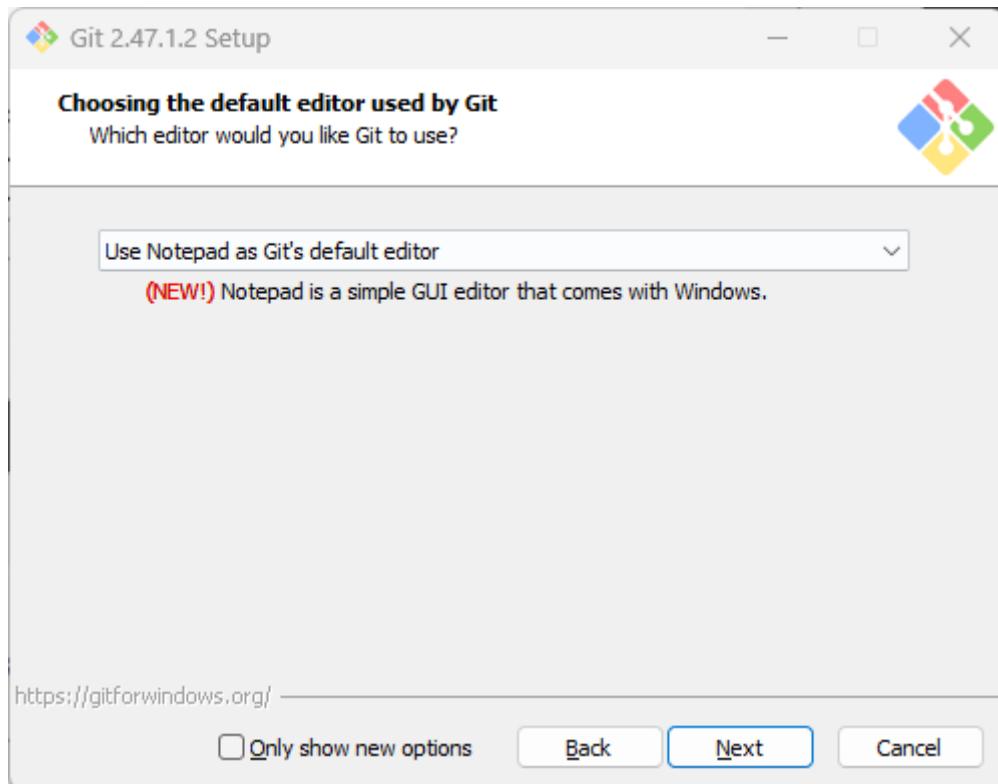
Y seguimos el asistente de instalación:



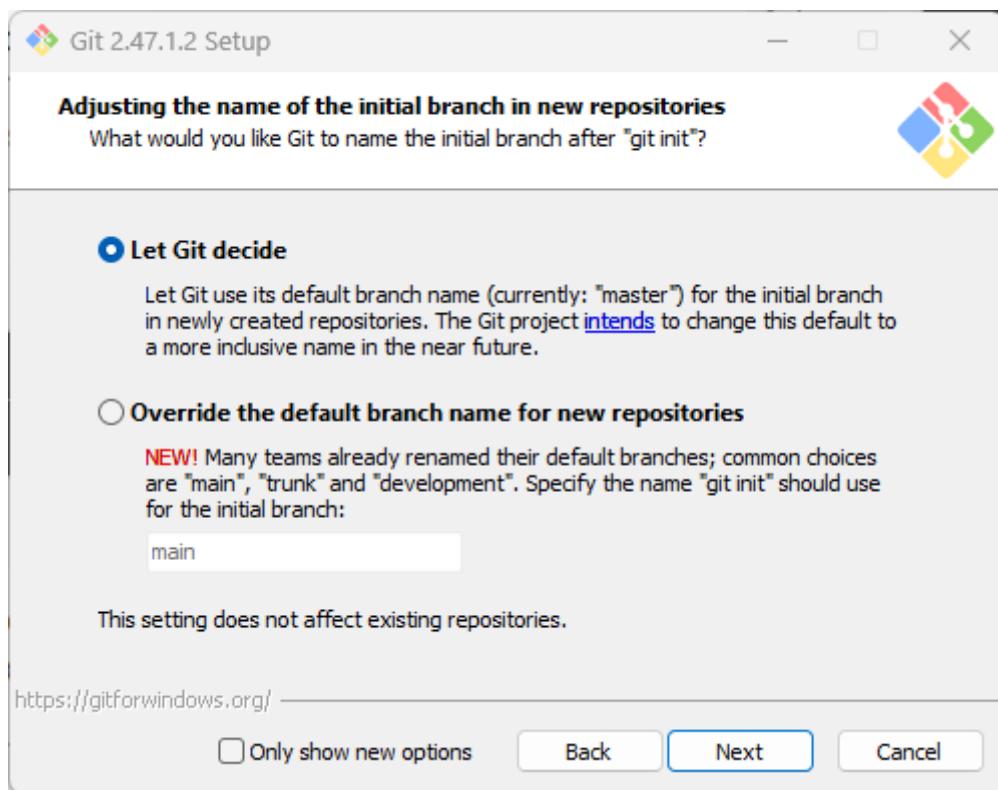
next:



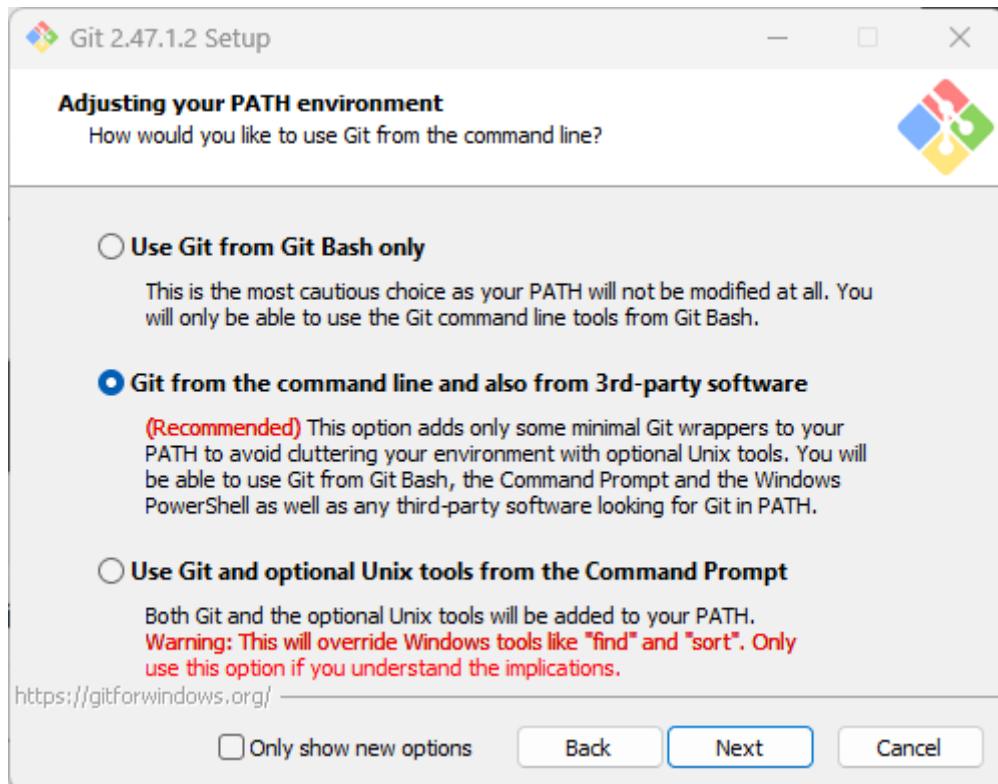
Next:



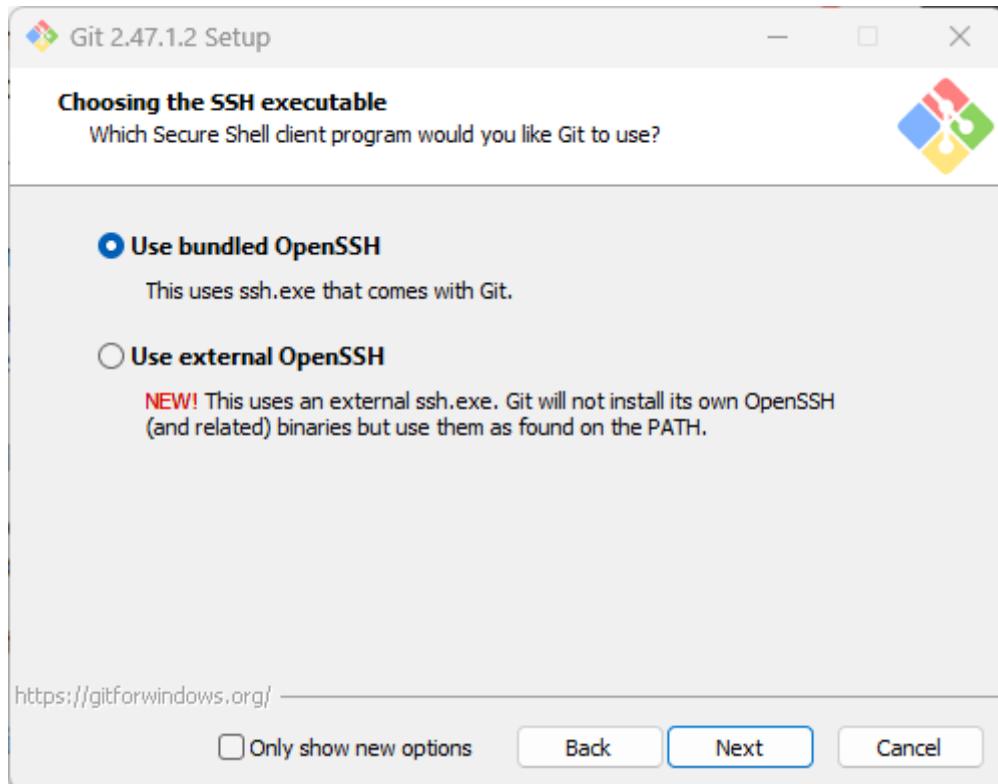
Next:



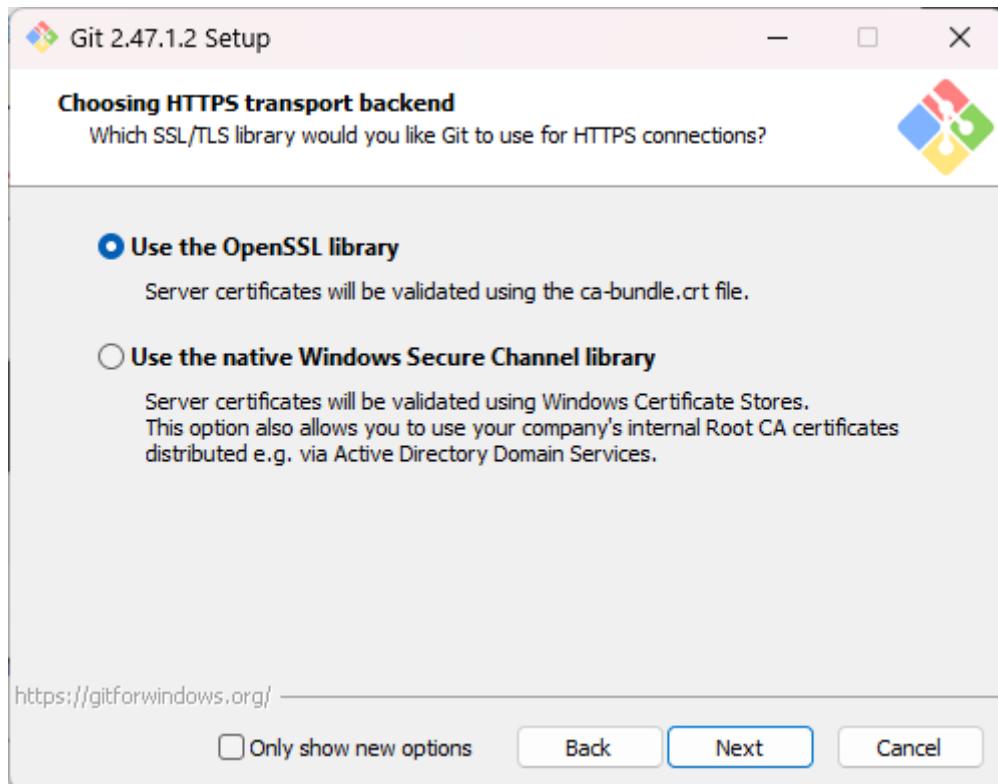
Next:



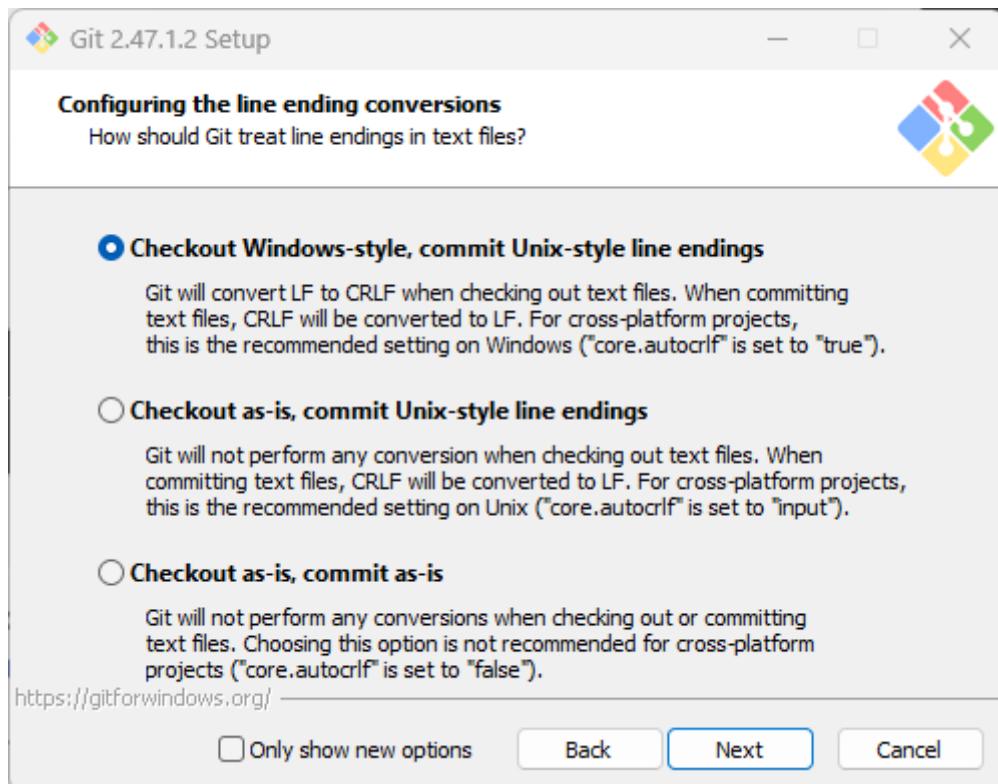
Next:



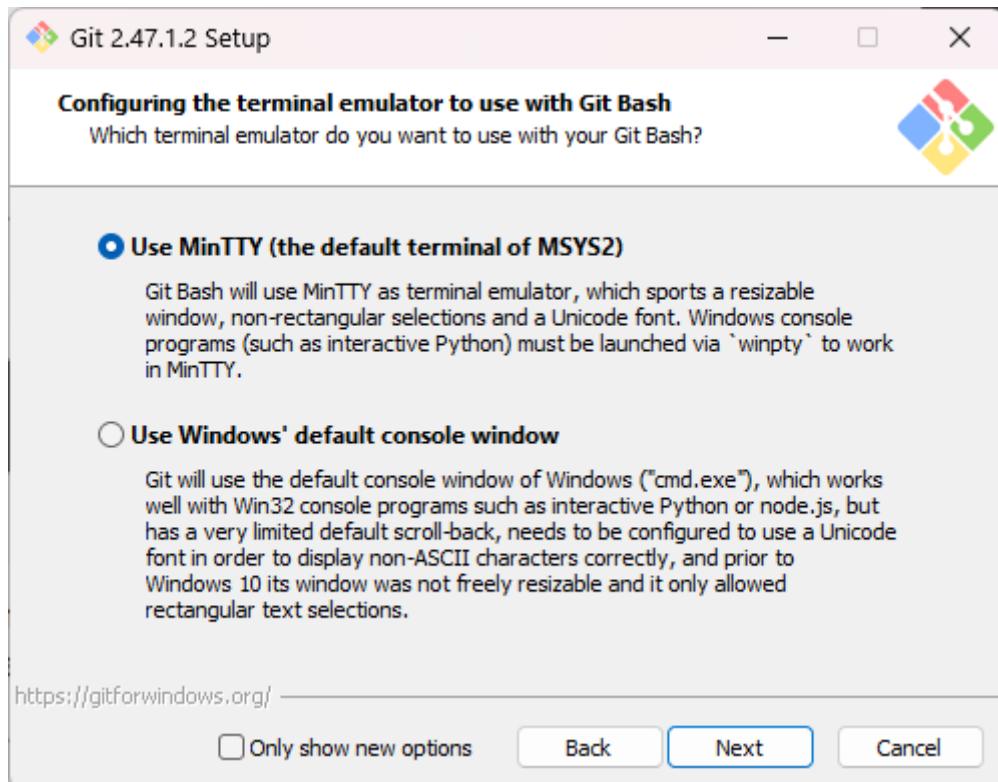
Next:



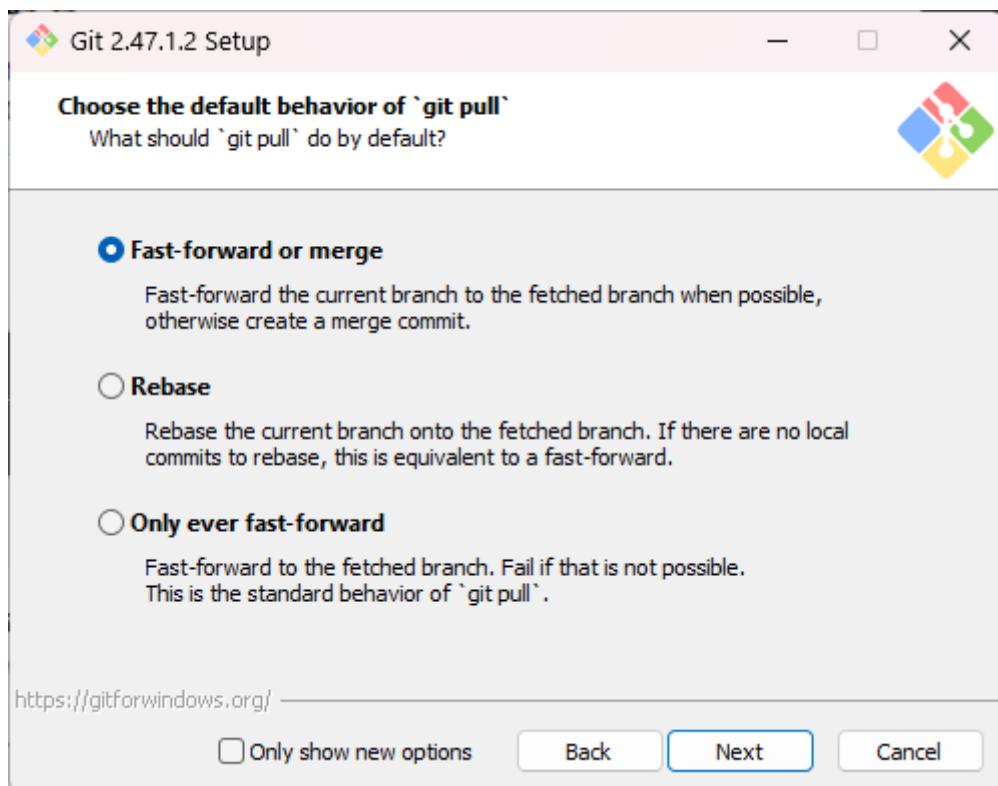
Next:



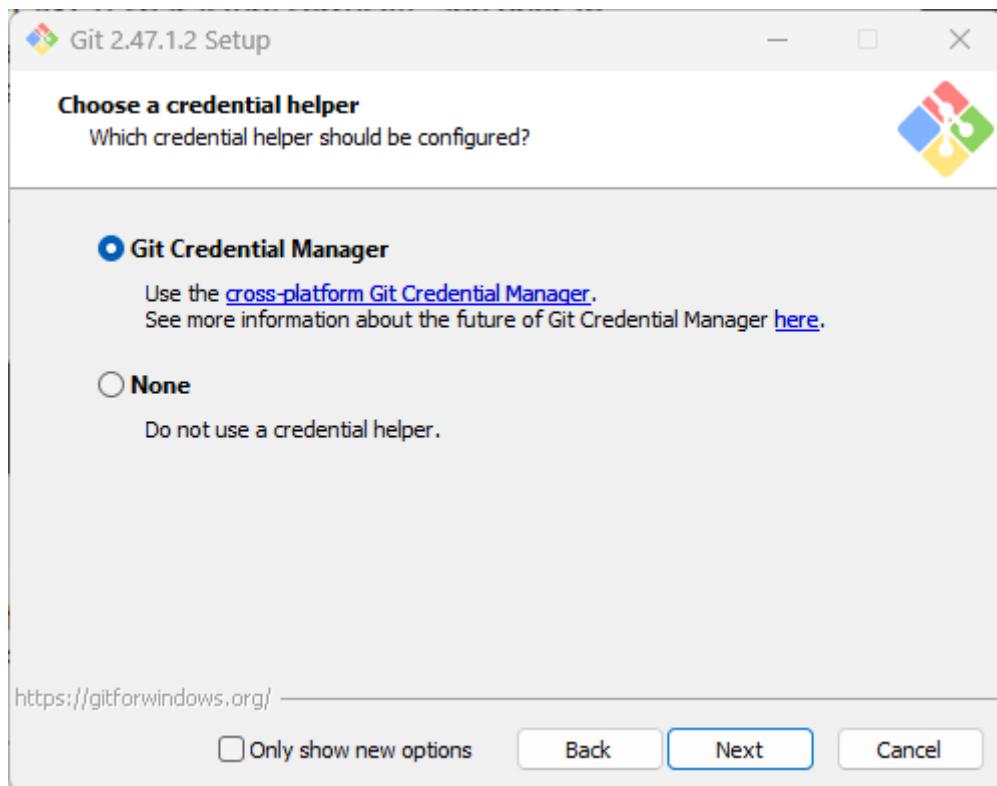
Next:



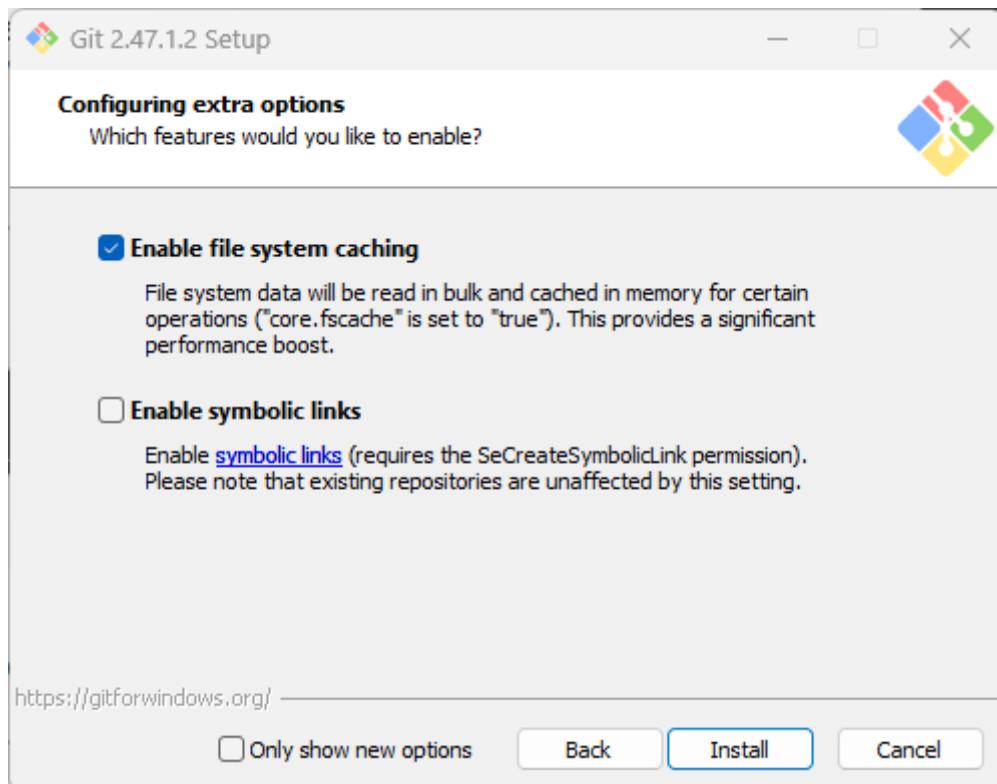
Next:



next:



Finalmente clic en install:



5. (RA6_f) Se ha garantizado la accesibilidad y seguridad de la

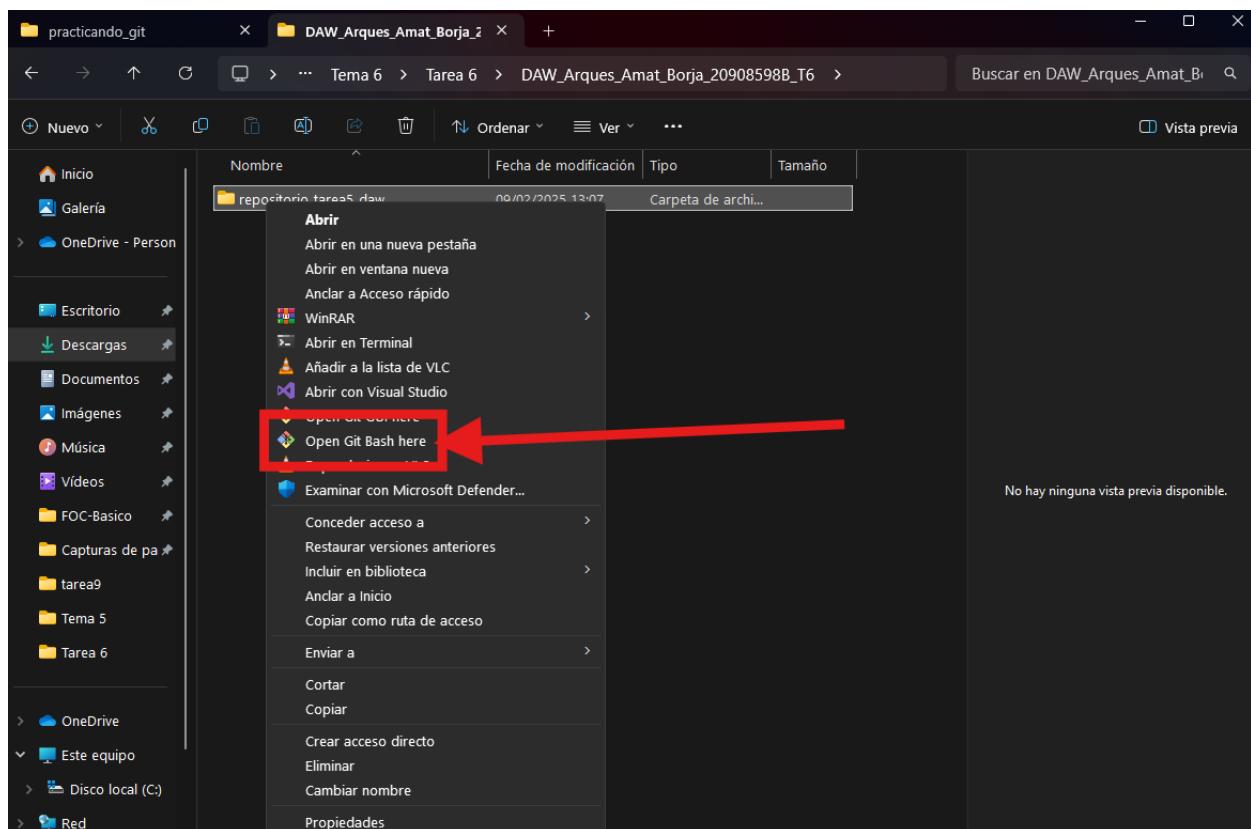
documentación almacenada por el sistema de control de versiones.

Una vez nos hemos creado una cuenta en GitHub y hemos instalado git en nuestra máquina, ya podemos empezar a crear nuestro repositorio local y conectarlo al remoto de GitHub. A continuación voy a explicar un poco como hacer los primeros pasos para conectarlos y la utilización de ciertos comandos que nos ayudaran a gestionar nuestro proyecto:

Podemos usar cualquier consola para manejar git desde nuestra máquina, pero a mí me gusta usar git bash.

Creamos una carpeta que será nuestro repositorio local que más tarde lo adjuntaremos a nuestro repositorio remoto:

Clic derecho sobre la carpeta y clic en git bash:



Nota: Yo ya tengo configurado git y GitHub con mi correo personal/principal/profesional, con lo que la cuenta creada ha sido simplemente una simulación para la tarea al igual que la instalación de git en mi máquina. Simplemente he descrito los pasos a seguir, por eso la cuenta que voy a usar ahora no es la misma que he creado en el primer punto de la tarea. Independientemente voy a explicar el tema de la seguridad SSH

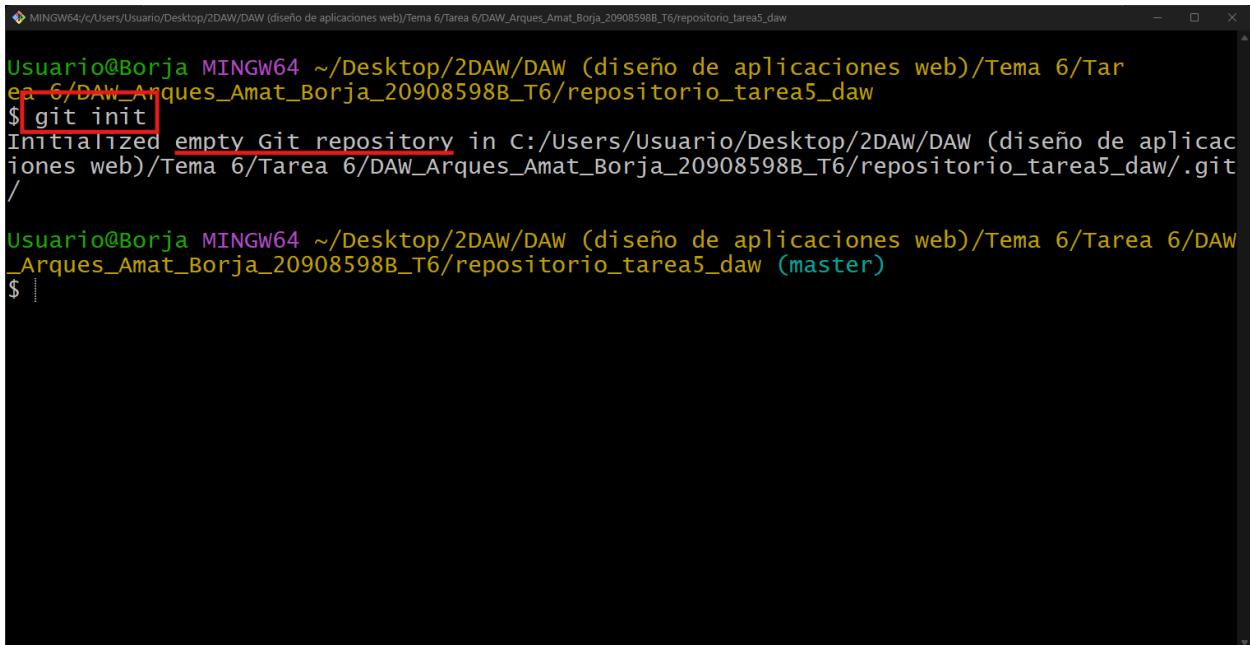
Se nos abrirá una terminal.

A screenshot of a terminal window titled "MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw". The window shows a command line interface with the user "Usuario@Borja" and the path to a directory. A single dollar sign (\$) is visible at the bottom left, indicating where input can be entered.

Lo primero sería inicializar el repositorio local en esta carpeta:

comando : (git init):

Y nos dice que se ha inicializado un repositorio vacío en la ruta especificada:



```
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw
$ git init
Initialized empty Git repository in C:/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw/.git/
/
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$
```

Ahora voy a explicar como hacer que nuestro Ordenador y GitHub se comuniquen:

Primero usamos:

Comandos:

```
git config --global user.name "usuario"
```

```
git config --global user.email "email@google.com"
```

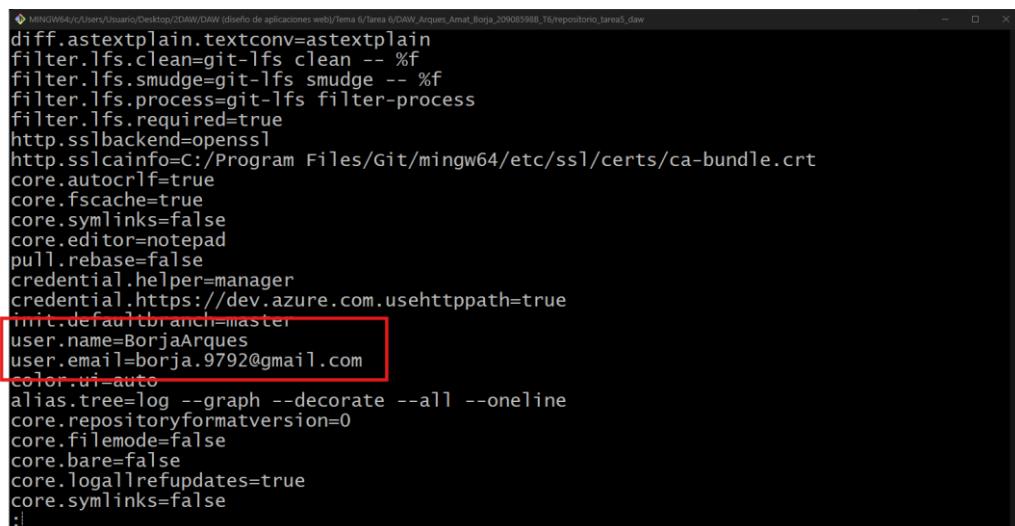


```
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ git config --global user.name "usuario"
git config --global user.email "email@gmail.com"
```

Luego listamos a ver si se ha agregado a la configuración:

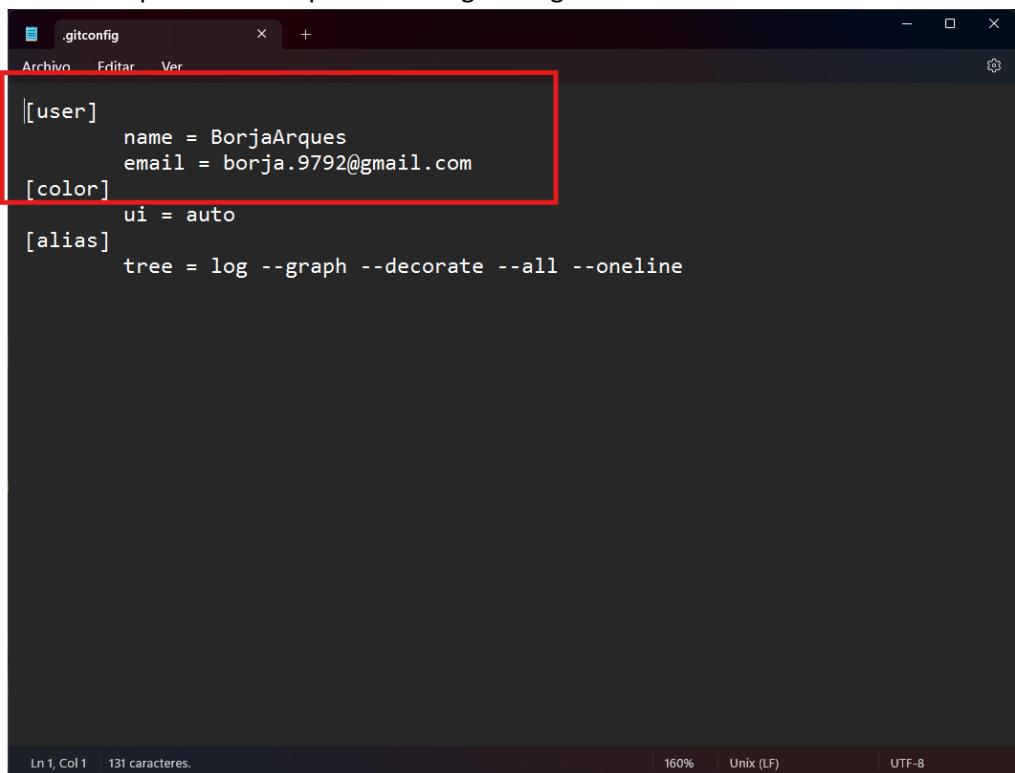
comando :

```
git config --list
```



```
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.editor=notepad
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=BorjaArques
user.email=borja.9792@gmail.com
color.ui=auto
alias.tree=log --graph --decorate --all --oneline
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
:|
```

Tambien lo podemos comprobar en el gitconfig



```
[user]
  name = BorjaArques
  email = borja.9792@gmail.com
[color]
  ui = auto
[alias]
  tree = log --graph --decorate --all --oneline
```

Siguiente paso sería configurar una Autentificación por SSH:

Adjunto la documentación oficial:

<https://docs.github.com/es/authentication/connecting-to-github-with-ssh/about-ssh>

Lo primero sería generar una SSH con el comando ::

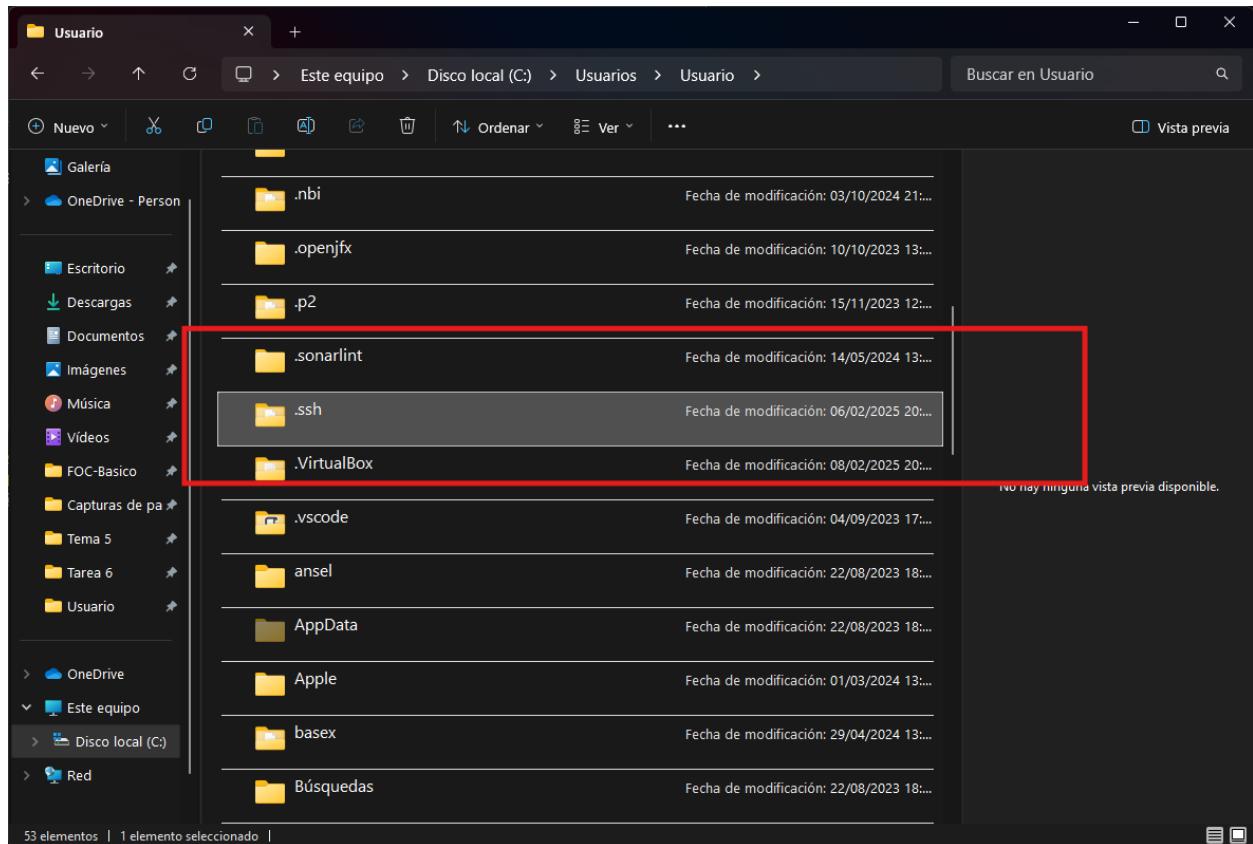
ssh-keygen -t ed25519 -C "Tuemail@ejemplo.com"

Ed25519 es el tipo de encriptación, en clase se usa rsa, yo voy a usar la que recomienda la documentación de GitHub:

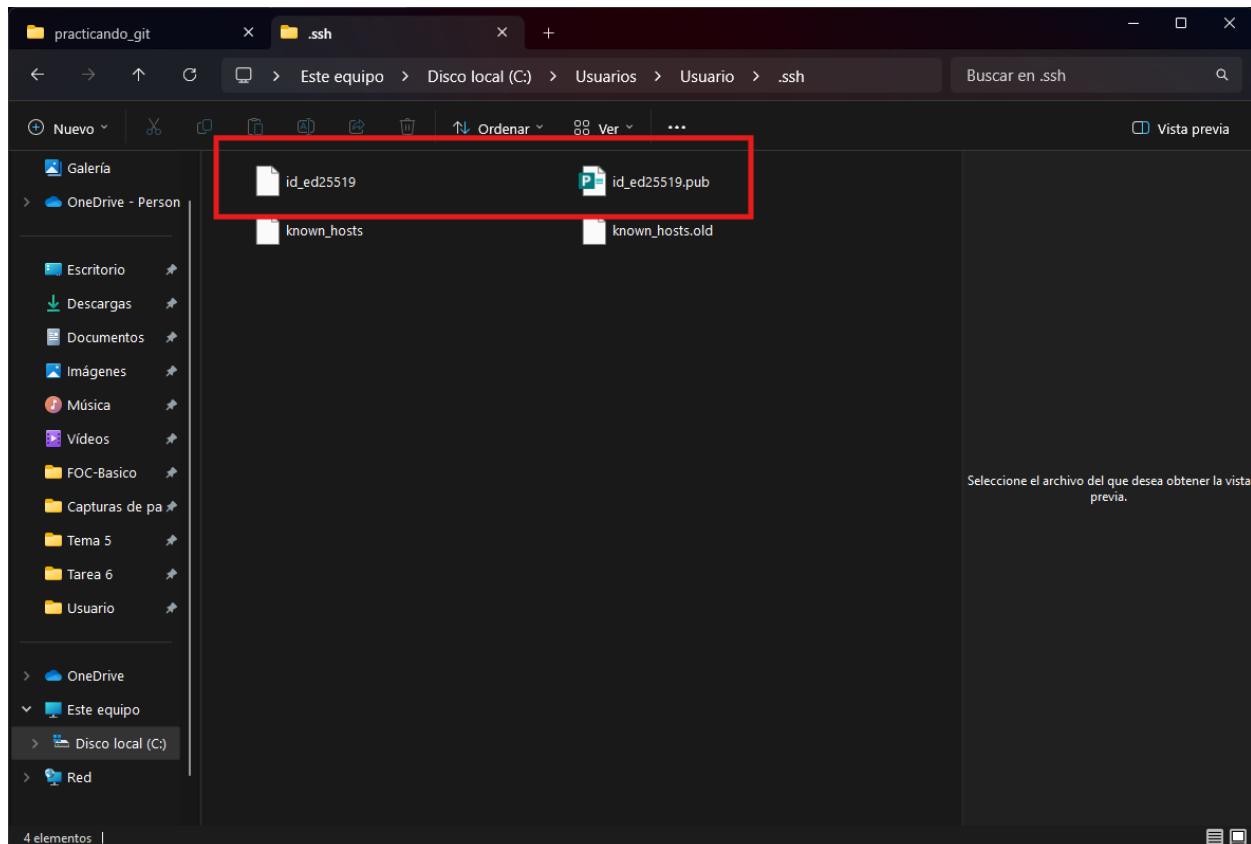


```
MINGW64/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ ssh-keygen -t ed25519 -C "borja.9792@gmail.com"
```

Yo ya tengo la mia creada, a continuación te pide que le digas donde guardar el archivo yo puse en su momento:



Id_ed25519 que seria la clave de conexión y id_ed25519.pub que seria la publica:



Para la tarea:

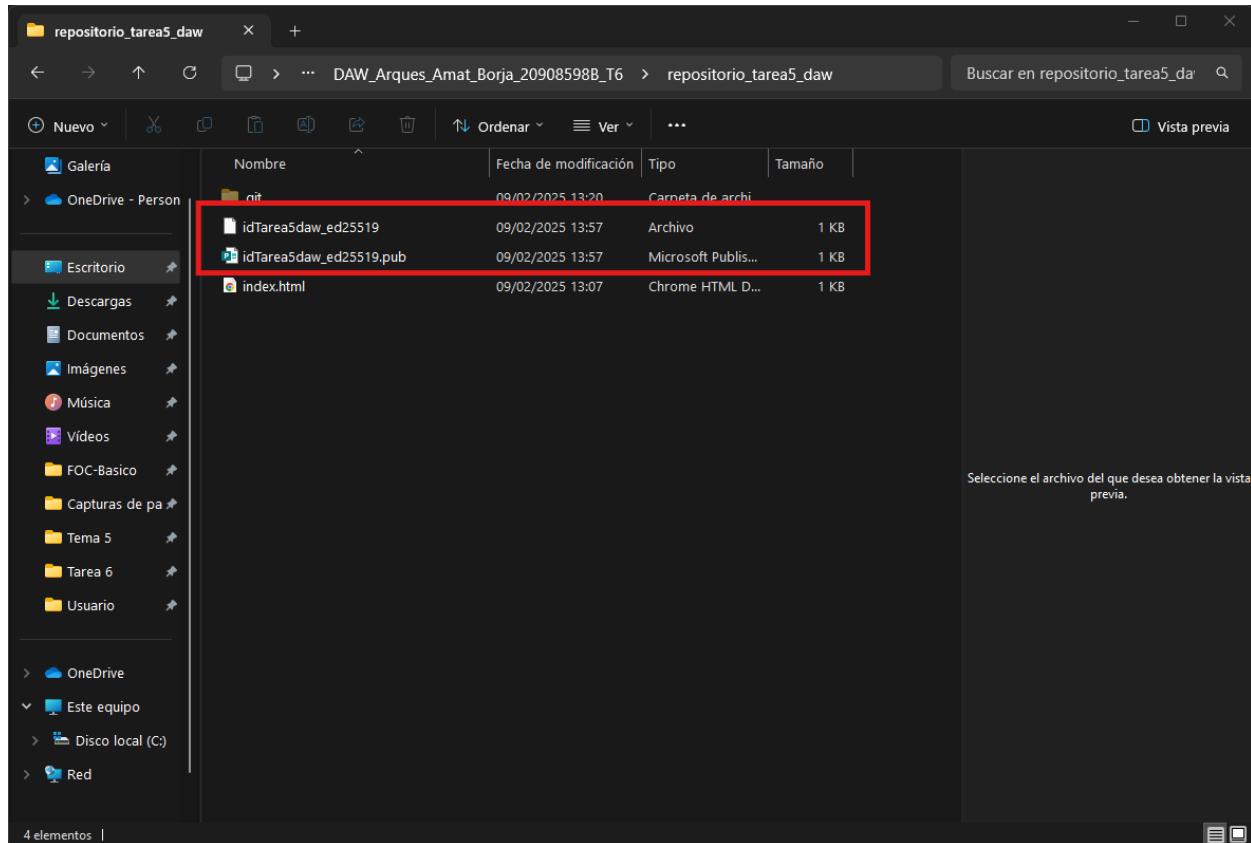
```

MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW
Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ ssh-keygen -t ed25519 -C "borja.9792@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Usuario/.ssh/id_ed25519): idTarea5daw_ed25
519
Enter passphrase for 'idTarea5daw_ed25519' (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in idTarea5daw_ed25519
Your public key has been saved in idTarea5daw_ed25519.pub
The key fingerprint is:
SHA256:rvMOEZcmsoZdAlM6mtWc/6oIZ0h+7rmy28JOU2xQPRc borja.9792@gmail.com
The key's randomart image is:
++-[ED25519 256]-
|   . . E.
| = O O .
| * =     O
| + + +
| o   + O S
| = O B .
| ..X B o
| o+X.*.
| .=BO=o
+---[SHA256]---

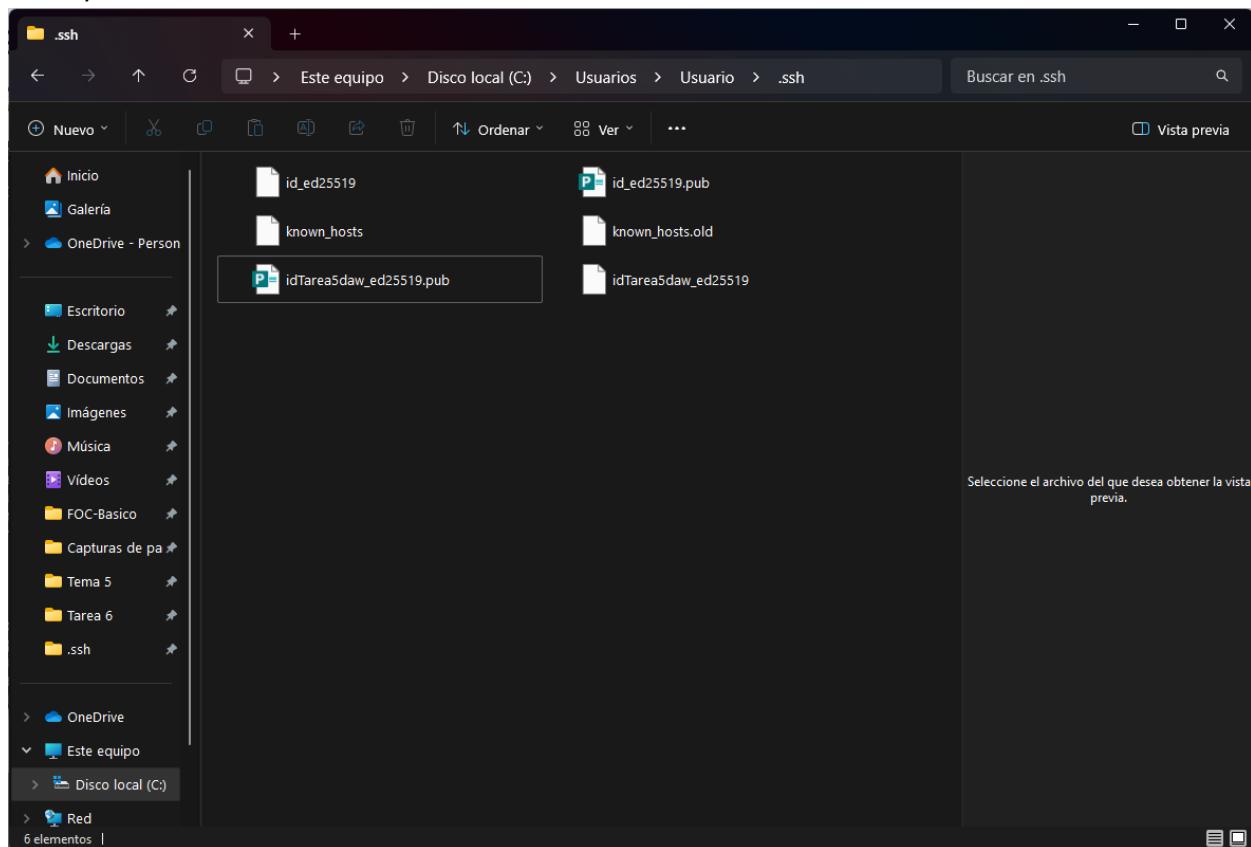
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW
Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ ...

```

Las he almacenado para la tarea en el documento de la tarea:



Las voy a mover manualmente donde están mis otras claves:



Una vez generadas tenemos que ir a GitHub:

The screenshot shows the GitHub Dashboard interface. On the right side, a vertical sidebar displays the user's profile information: **borjadaw92**, **Set status**, **Your profile**, **Your repositories**, **Your Copilot**, **Your projects**, **Your stars**, **Your gists**, **Your organizations**, **Your enterprises**, **Your sponsors**, **Try Enterprise** (Free), **Feature preview** (New), **Settings** (highlighted with a red arrow), **GitHub Website**, **GitHub Docs**, **GitHub Support**, and **GitHub Community**. Below this sidebar, there are links for **Sign in** and **Sign out**.

Top repositories section shows **tmux / tmux** (16.4k stars) and **curl / curl** (36.7k stars).

Home section includes **Introduction to GitHub**, **GitHub Pages**, **Code with Copilot**, and **Hello GitHub Actions**.

Start a new repository for borjadaw92 form shows a **Repository name *** field with "name your new repository..." and a **Visibility** switch between **Public** and **Private**.

Introduce yourself with a profile README section shows a sample README file content:

```

1 - 🌟 Hi, I'm @borjadaw92
2 - 🌐 I'm interested in ...
3 - 🚗 I'm currently learning ...
4 - 💡 I'm looking to collaborate on ...
5 - 📩 How to reach me ...
6 - 🎉 Pronouns: ...

```

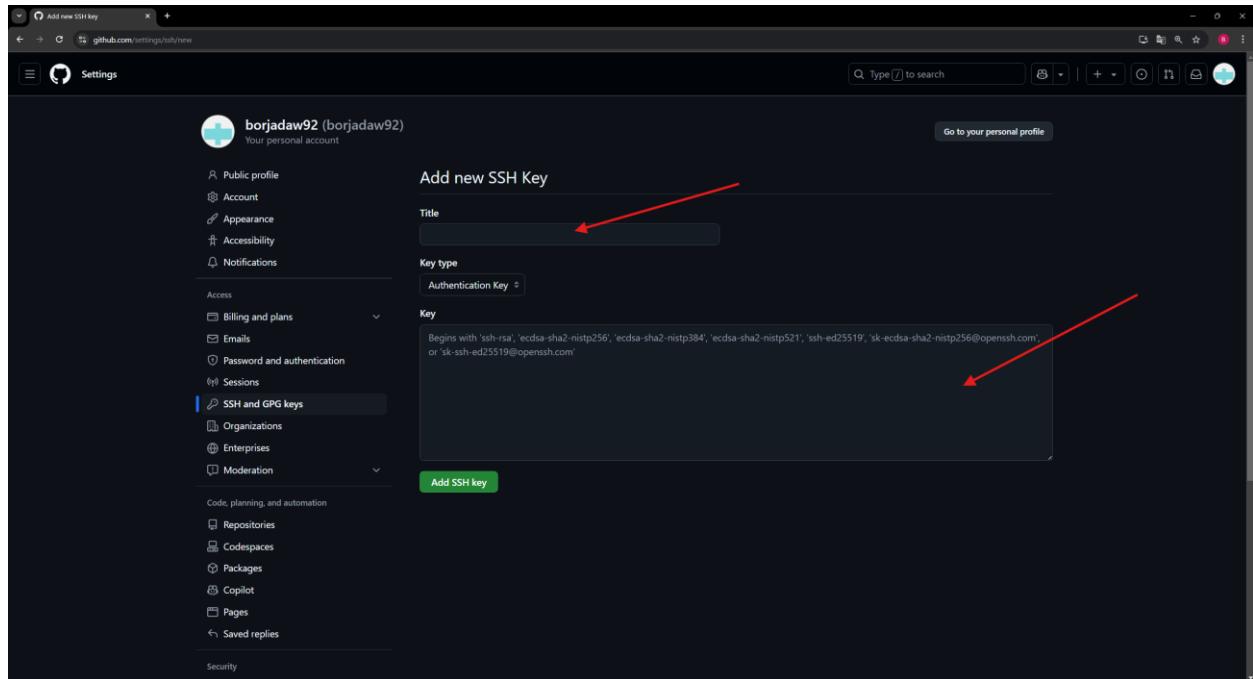
A red arrow points from the top right towards the **Ask Copilot** button.

This screenshot shows the GitHub 'Public profile' settings page. The left sidebar has a 'SSH and GPG keys' section highlighted with a red arrow. The main area shows fields for Name, Profile picture, Bio (with a note about verified email), Pronouns, URL, ORCID ID, and Social accounts.

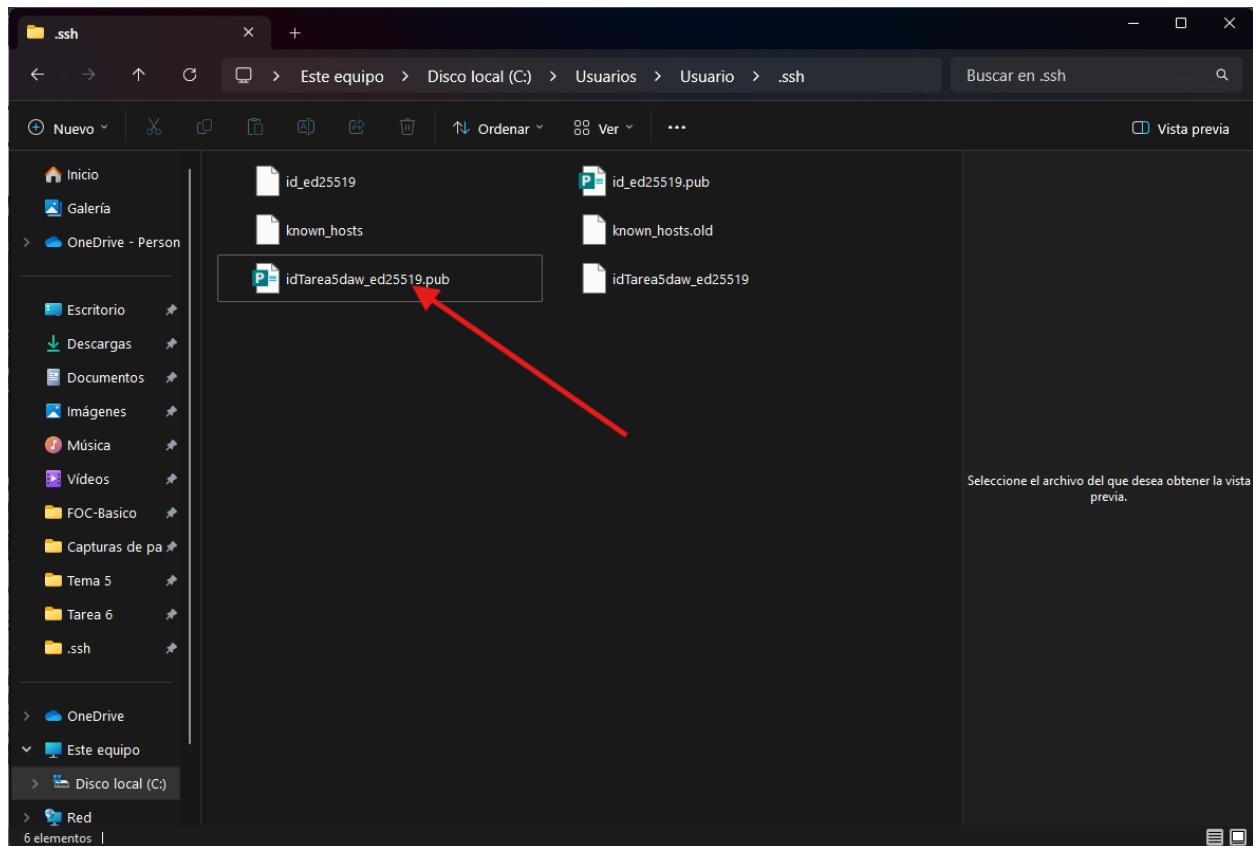
This screenshot shows the GitHub 'SSH and GPG keys' settings page. The left sidebar has a 'SSH and GPG keys' section highlighted with a red arrow. The main area shows sections for SSH keys and GPG keys, with a 'New SSH key' button highlighted with a red arrow.

Una vez nos hemos movido por los menus de GitHub hasta donde he indicado en la imagen anterior,

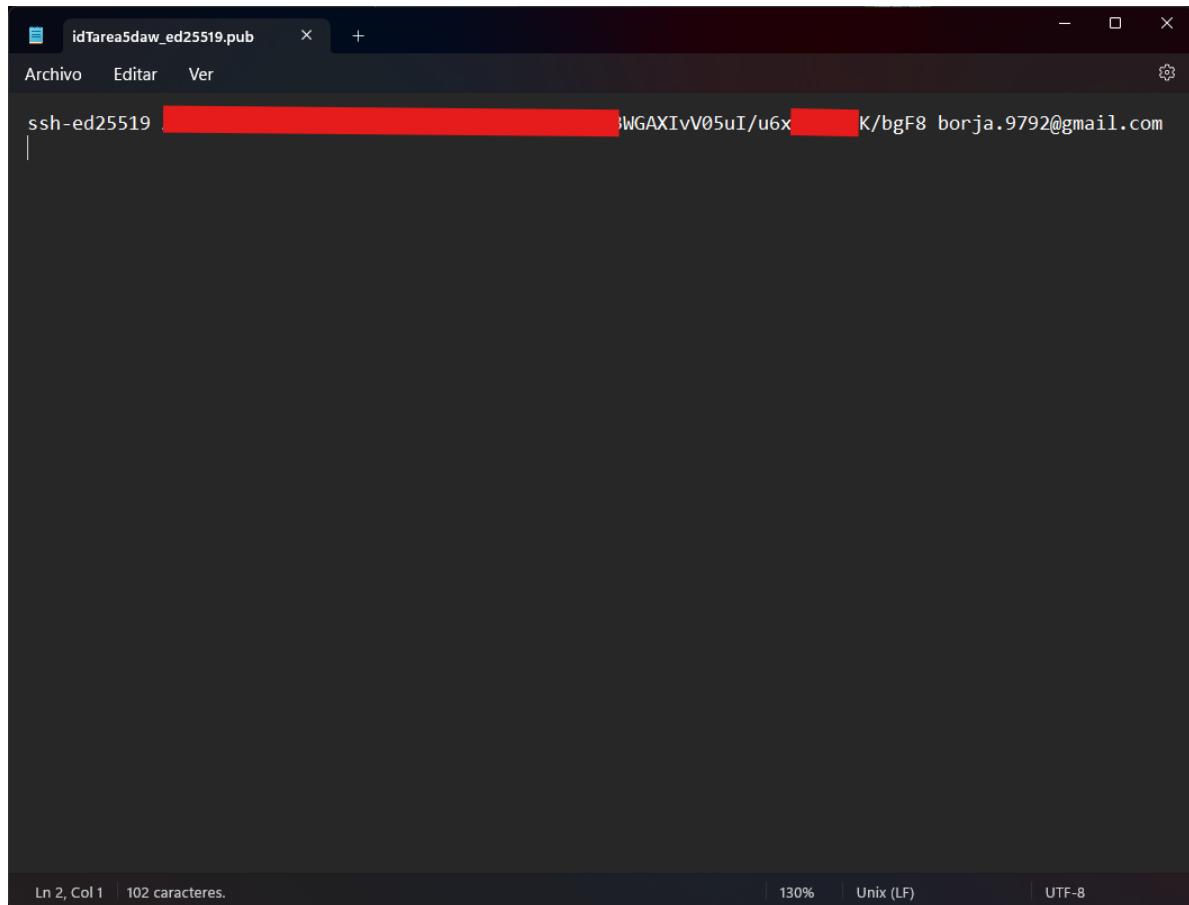
debemos llenar los datos , el nombre , el que queramos que es para la tarea 6 de daw:



Y la Key es la encriptación generada que encontramos en nuestro documento IMPORTANTE: PUBLICO, el .pub, no el privado//original si lo abrimos con txt por ejemplo:



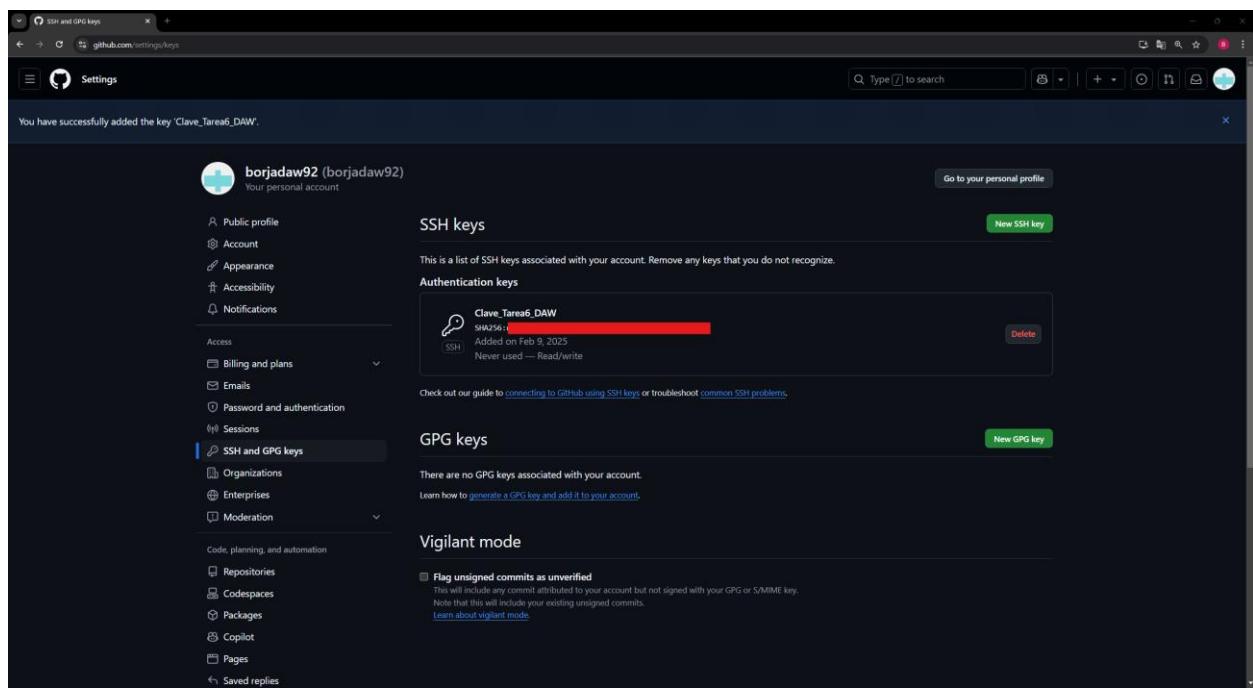
Tapo parte de la clave por seguridad, pero debemos copiar todo eso y pegarlo en el text área de GitHub:



```
ssh-ed25519 [REDACTED] !WGAXIVV05uI/u6x [REDACTED] K/bgF8 borja.9792@gmail.com
```

Ln 2, Col 1 102 caracteres. 130% Unix (LF) UTF-8

Le damos a ADD y ya tendríamos nuestra clave vinculada:



You have successfully added the key 'Clave_Tarea6_DAW'.

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

 Clave_Tarea6_DAW	SHA256: [REDACTED]	Added on Feb 9, 2025	Delete
SSH			
Never used — Read/write			

[Check out our guide to connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

There are no GPG keys associated with your account.

[Learn how to generate a GPG key and add it to your account](#).

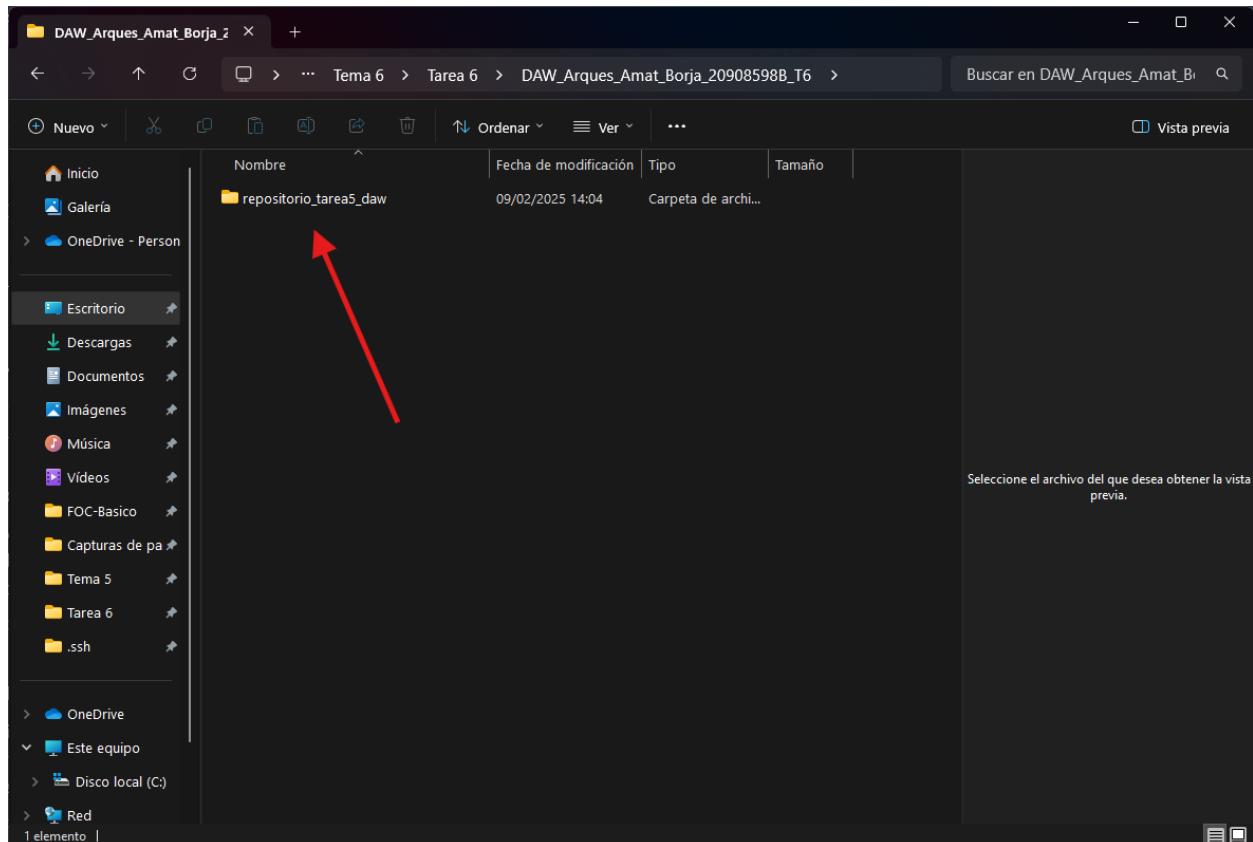
Vigilant mode

[Flag unsigned commits as unverified](#)

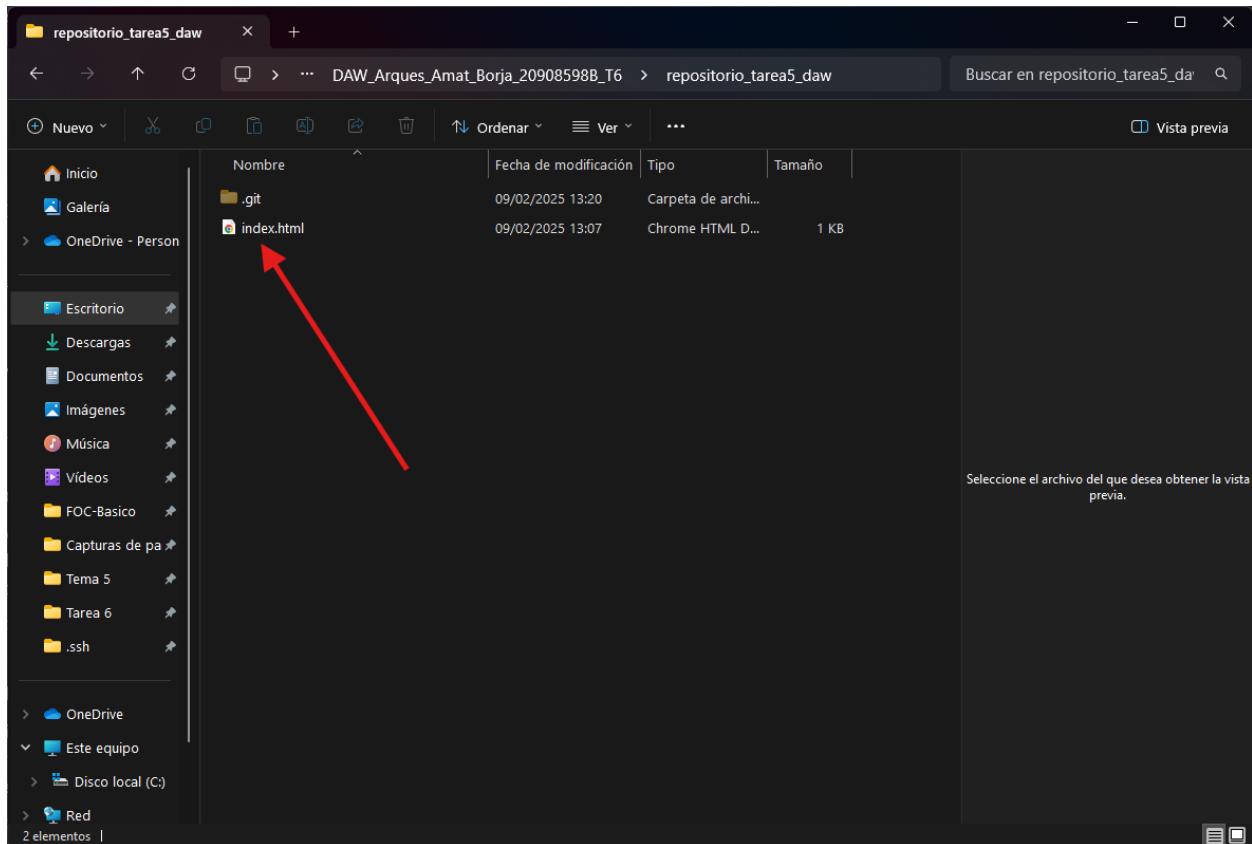
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
[Learn about vigilant mode](#)

Como he explicado antes, estos pasos son "simulados" ya que yo ya tenía todas esta configuración hecha en mi cuenta personal. Así que voy a ponerme en situación:

Tengo repositorio local creado:



me descargo el index.html para la tarea que proporciona el profesor y lo meto dentro del repositorio local:



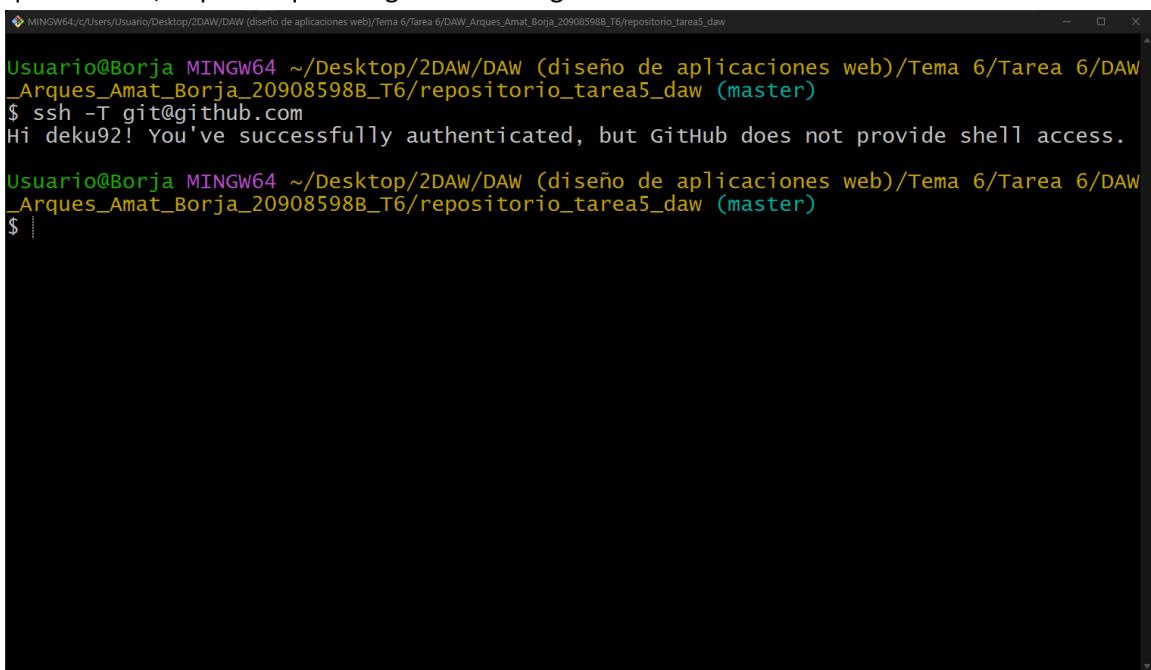
y tengo la cuenta GitHub con una clave SSH configurada:

The screenshot shows the GitHub 'SSH keys' settings page. On the left, there's a sidebar with options like 'Public profile', 'Account', 'Appearance', 'Accessibility', 'Notifications', 'Access', 'Billing and plans', 'Emails', 'Password and authentication', 'Sessions', 'SSH and GPG keys' (which is selected), 'Organizations', 'Enterprises', and 'Moderation'. The main content area is titled 'SSH keys' and contains the message: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' It lists one key: 'Clave_acceso' (SSH icon, added on Feb 6, 2025, last used within the last week, Read/write permissions). There are 'New SSH key' and 'Delete' buttons. Below this is a 'GPG keys' section with the message: 'There are no GPG keys associated with your account.' It has a 'New GPG key' button. At the bottom is a 'Vigilant mode' section.

Por último, con el comando ssh -T gitgithub.com

Le vamos a decir que nos recuerde para no estar pidiendo claves constantemente cada vez que quieras aplicar una subida a nuestro proyecto:

Nota: Como ya lo tenia configurado no me sale el mensaje de confirmación , pero si es la primera vez que lo haces, te pedirá que le digas si estas seguro:



```
MINGW64/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ ssh -T git@github.com
Hi deku92! You've successfully authenticated, but GitHub does not provide shell access.

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$
```

Agrego este paso de arriba pero la documentación oficial de GitHub a mi me da el mensaje de bienvenida directamente por que ya lo tenía configurado:

- 2 Escriba lo siguiente:

```
Shell
ssh -T git@github.com
# Attempts to ssh to GitHub
```

Puedes ver una advertencia como la siguiente:

```
> The authenticity of host 'github.com (IP ADDRESS)' can't be established.
> ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
> Are you sure you want to continue connecting (yes/no)?
```

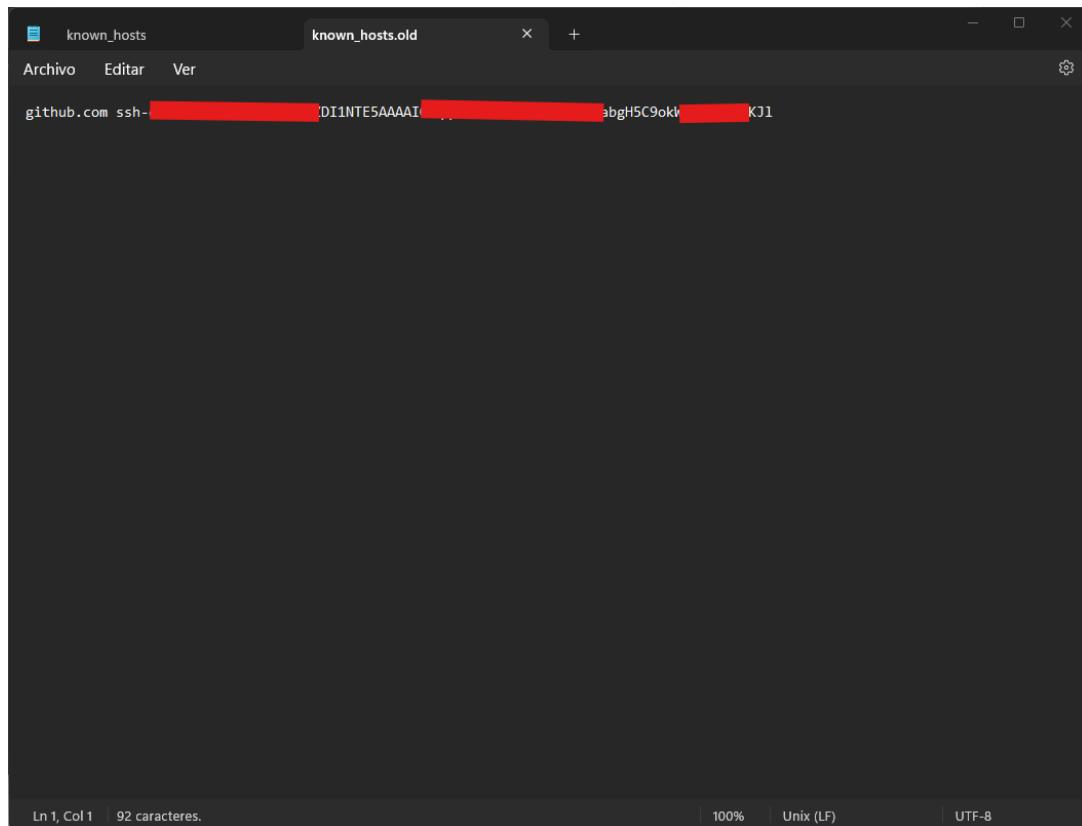
- 3 Compruebe que la huella digital del mensaje que ve coincide con [la huella digital de clave pública de GitHub](#). En caso afirmativo, escriba `yes`:

```
> Hi USERNAME! You've successfully authenticated, but GitHub does not
> provide shell access.
```

Note

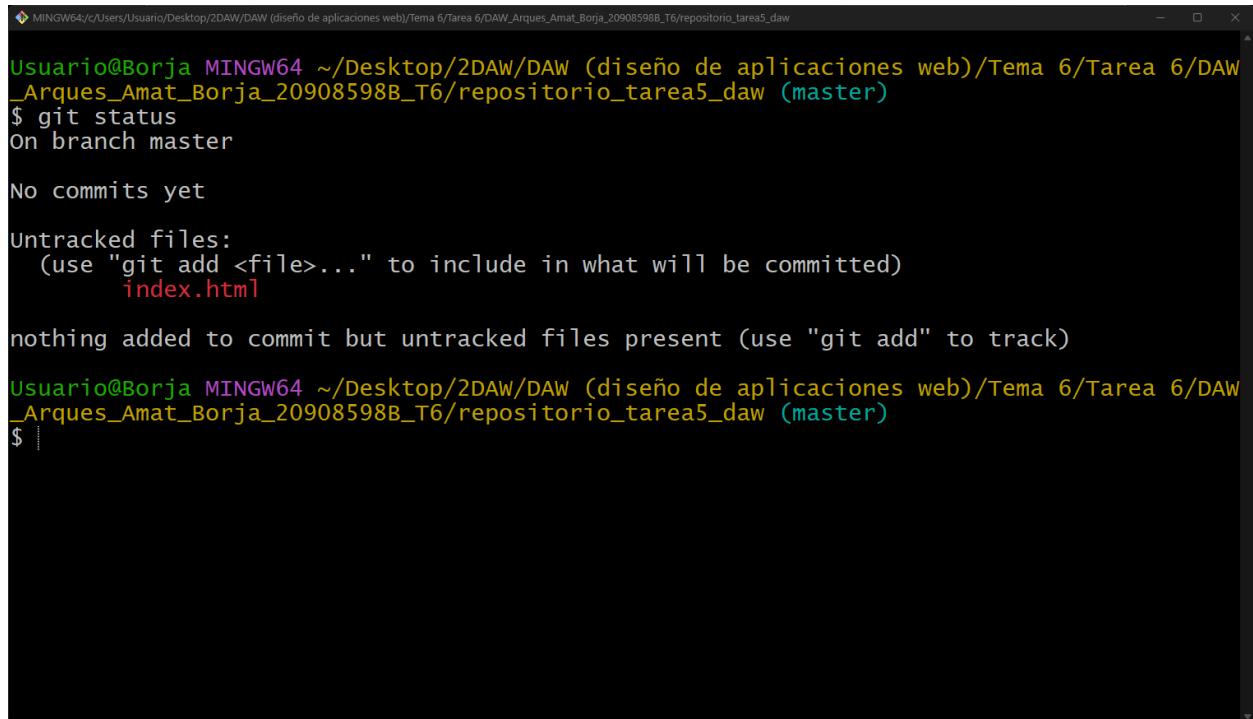
El comando remoto debe cerrarse con el código 1.

Cuando usamos ese comando se nos genera un documento llamado known_hosts, ahí también podemos comprobar si se ha agregado github con nuestra clave SSH:



A screenshot of a terminal window titled "known_hosts" with the tab "known_hosts.old". The window has a dark background. At the top, there are menu options: Archivo, Editar, Ver. Below the menu, there is a single line of text: "github.com ssh-... DI1NTE5AAAAI... bghH5C9ok... KJ1". At the bottom of the terminal window, there is status information: "Ln 1, Col 1 | 92 caracteres.", "100%", "Unix (LF)", and "UTF-8".

Si hago un git status me dice que tengo index.html y que no he hecho nada con el todavía:



```

MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW
_ Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ git status
On branch master

No commits yet

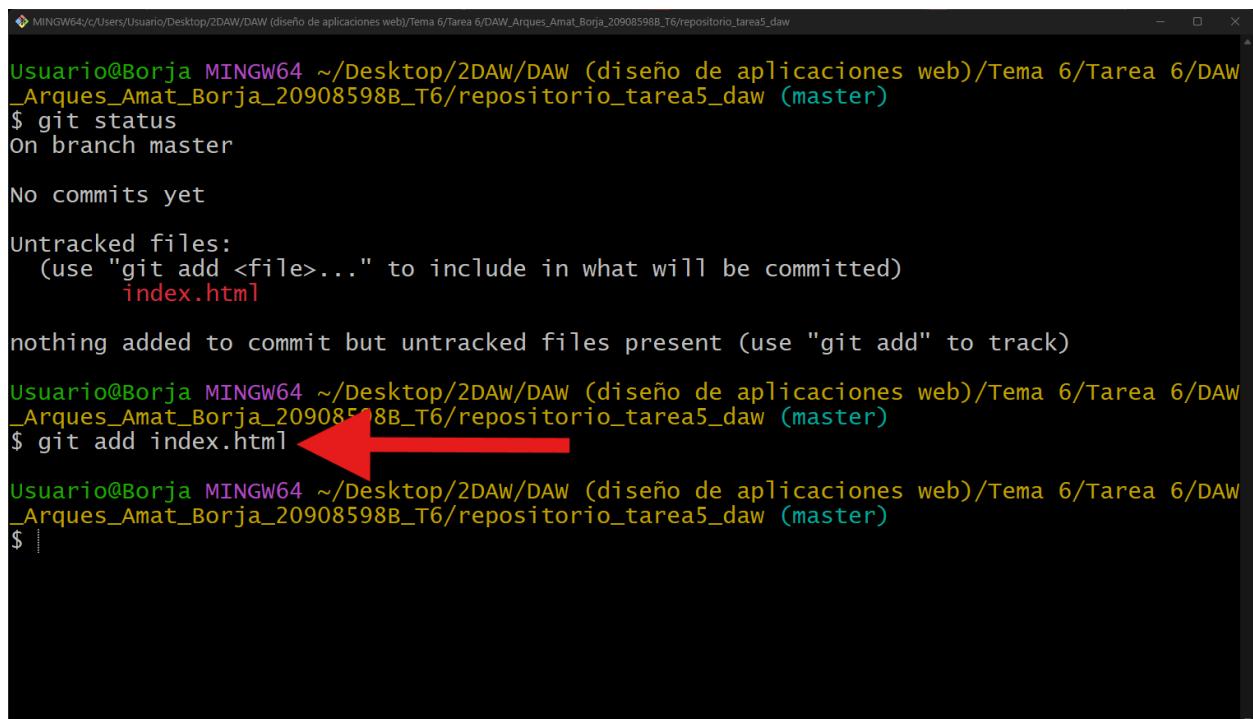
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html

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

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW
_ Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ ...

```

Lo añadimos con git add . (para todo lo que este pendiente) o git add “nombre del archivo” (para solo añadir el documento que queramos):



```

MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW
_ Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ git status
On branch master

No commits yet

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

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

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW
_ Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ git add index.html ←
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW
_ Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ ...

```

Ahora ya detecta que el index.html esta listo para hacerle commit:

```

MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ git status
On branch master

No commits yet

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

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea5_daw (master)
$ ...

```

Hacemos nuestro primer git commit – m “comentario para identificar el commit/versión de nuestro proyecto”:

```

MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git status
On branch master

No commits yet

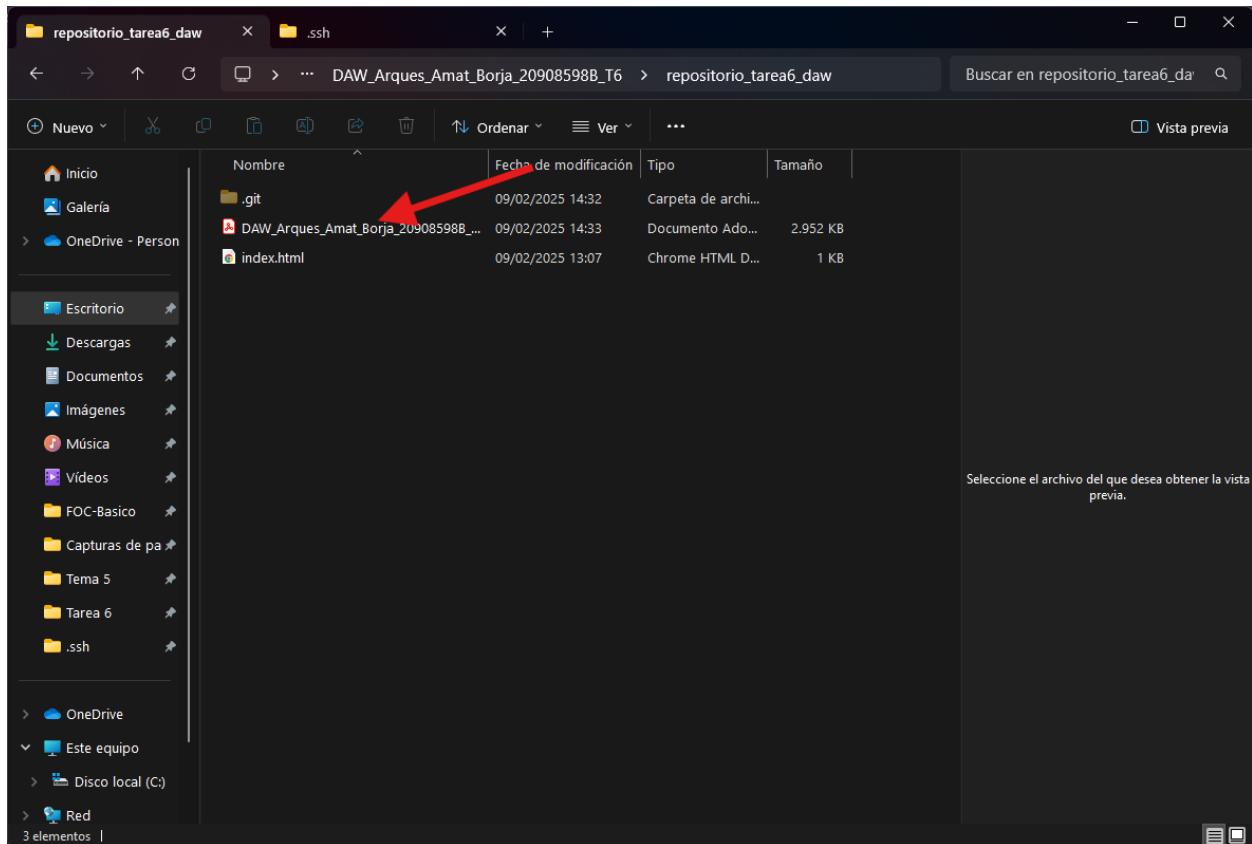
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git commit -m "Primer commit tarea6 DAW"
[master (root-commit) e346f49] Primer commit tarea6 DAW
 1 file changed, 15 insertions(+)
 create mode 100644 index.html

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ ...

```

Voy a hacer guardar “la plantilla de la tarea por primera vez dentro del repositorio”:



hago un status de mi local:

```

MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw
g
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ i
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    DAW_Arques_Amat_Borja_20908598B_T6.pdf

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

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ ...

```

lo añado con git add y se queda a la espera de que le haga commit el cual seria mi segundo commit registrado:

```

$ i
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    DAW_Arques_Amat_Borja_20908598B_T6.pdf

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

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git add .

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   DAW_Arques_Amat_Borja_20908598B_T6.pdf

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ 

```

commit – m

```

$ MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw
8598B_T6/repositorio_tarea6_daw (master)
$ git add .

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   DAW_Arques_Amat_Borja_20908598B_T6.pdf

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git commit -m "Segundo commit commit del pdf de la tarea6 DAW"
[master 034cbf3] Segundo commit commit del pdf de la tarea6 DAW
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 DAW_Arques_Amat_Borja_20908598B_T6.pdf

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git status
On branch master
nothing to commit, working tree clean

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ 

```

si tiro un status me dice que no hay nada por hacer commit:

```
MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw
8598B_T6/repositorio_tarea6_daw (master)
$ git add .

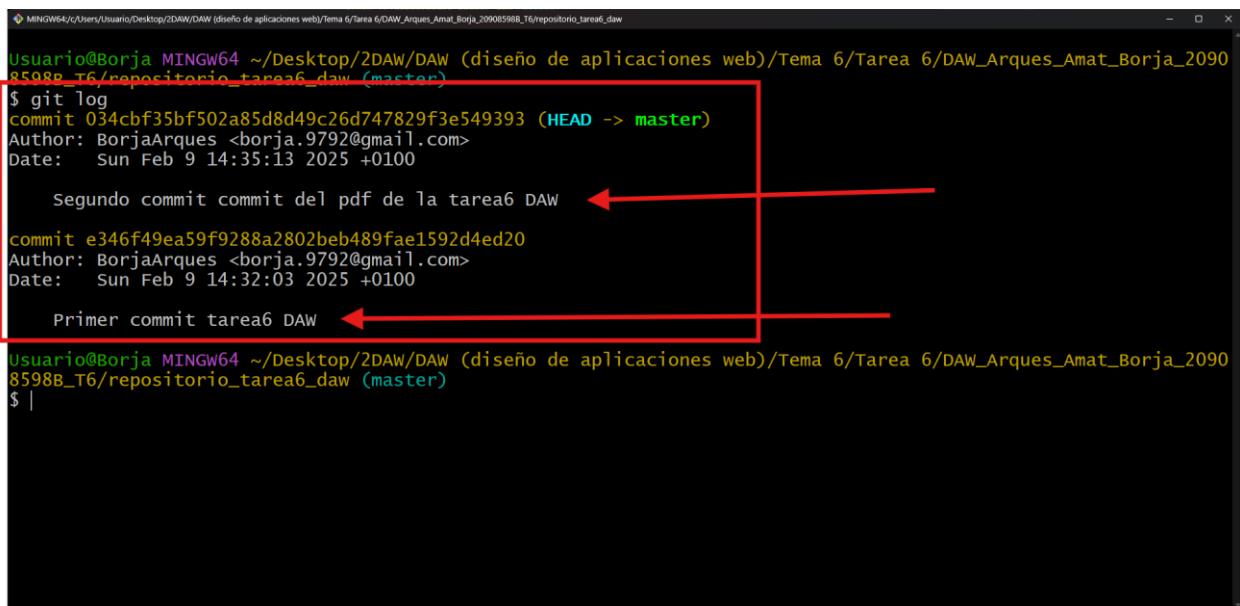
Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_2090
8598B_T6/repositorio_tarea6_daw (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   DAW_Arques_Amat_Borja_20908598B_T6.pdf

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_2090
8598B_T6/repositorio_tarea6_daw (master)
$ git commit -m "Segundo commit del pdf de la tarea6 DAW"
[master 034cbf3] Segundo commit del pdf de la tarea6 DAW
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 DAW_Arques_Amat_Borja_20908598B_T6.pdf

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_2090
8598B_T6/repositorio_tarea6_daw (master)
$ git status
On branch master
nothing to commit, working tree clean

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_2090
8598B_T6/repositorio_tarea6_daw (master)
$ |
```

para ver como va mi rama podemos hacer un git log donde vemos las modificaciones que se ha realizado en el repositorio y los comentarios de los commit:



```
MINGW64:/c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_2090
8598B_T6/repositorio_tarea6_daw (master)
$ git log
commit 034cbf35bf502a85d8d49c26d747829f3e549393 (HEAD -> master)
Author: BorjaArques <borka.9792@gmail.com>
Date:  Sun Feb 9 14:35:13 2025 +0100

  Segundo commit commit del pdf de la tarea6 DAW

commit e346f49ea59f9288a2802beb489fae1592d4ed20
Author: BorjaArques <borka.9792@gmail.com>
Date:  Sun Feb 9 14:32:03 2025 +0100

  Primer commit tarea6 DAW

Usuario@Borja MINGW64 ~/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_2090
8598B_T6/repositorio_tarea6_daw (master)
$ |
```

Para acabar la tarea realizare un tercer commit del pdf terminado, pero mientras escribo esto no tendré ese tercer commit hecho ya que el pdf no esta completo por eso igual se puede observar varios cambios diferentes a las capturas que aporto aquí.

La conexión con nuestro repositorio remoto se puede hacer antes o después de haber trabajado con el proyecto en local lo único que es indispensable es tener si se desea la clave SSH configurada como es el caso, repositorio local y repositorio remoto creados:

vamos a GitHub y copiamos el enlace que nos proporciona el repositorio con clave SSH:

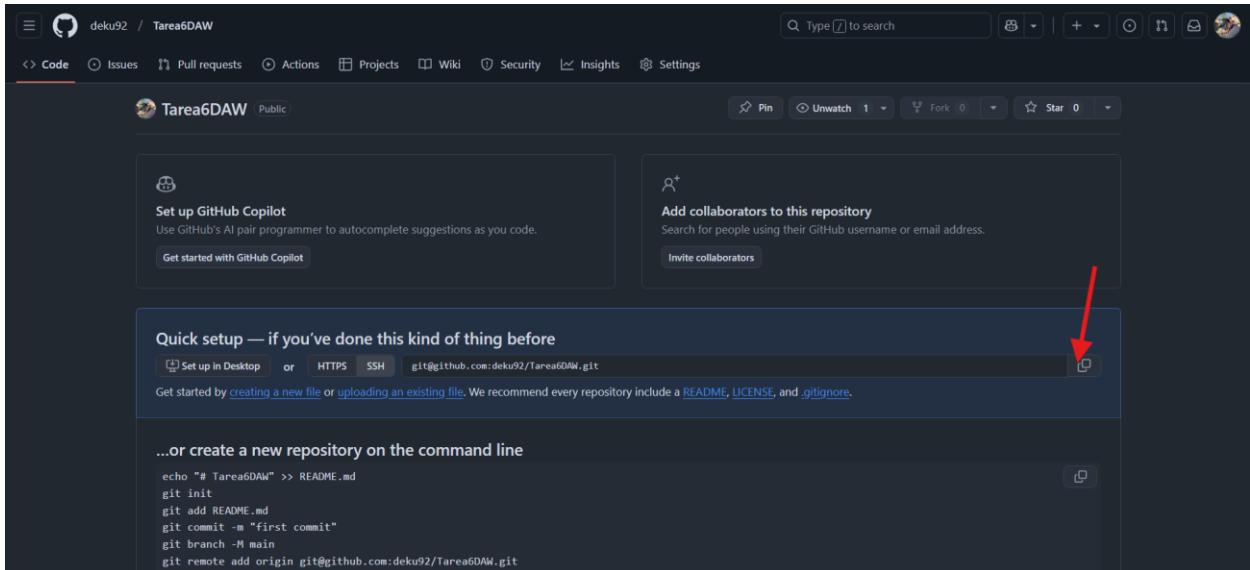
The screenshot shows the GitHub Home page. On the left, there's a sidebar with 'Top repositories' and a search bar. In the center, there's a 'Trending repositories' section with two items: 'dzhng/deep-research' and 'langgenius/dify'. On the right, there's a 'GitHub Copilot' section with a banner and a 'Latest changes' feed. A red arrow points from the text above to the 'Tarea6DAW' repository in the sidebar.

The screenshot shows the GitHub repository settings page for 'Tarea6DAW'. It has sections for 'Set up GitHub Copilot', 'Add collaborators to this repository', and a 'Quick setup' section. The 'Quick setup' section contains links for 'Set up in Desktop', 'HTTPS', and 'SSH'. A red arrow points to the 'SSH' link. Below it, there's a command-line guide for creating a repository.

```

echo "# Tarea6DAW" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:deku92/Tarea6DAW.git
  
```

Copiamos



nos vamos a git bash y aplicamos el comando git remote add origin "link SSH copiado"

Y estarán conectados pero aun no veremos ningún archivo en GitHub por que deberemos de hacer un push a los documentos commiteados anteriormente.

```

MINGW64 /c/Users/Usuario/Desktop/2DAW/DAW (diseño de aplicaciones web)/Tema 6/Tarea 6/DAW_Arques_Amat_Borja_20908598B_T6/repositorio_tarea6_daw (master)
$ git remote add origin git@github.com:deku92/Tarea6DAW.git ←

```

Antes de hacer el push voy a terminar la tarea:

-acabo el documento.

-guardo en PDF.

-hago un tercer commit con el pdf acabado de la tarea.

-hago push del “proyecto”.

6. (RA6_g) Se ha documentado la instalación, configuración y uso del sistema de control de versiones utilizado.