

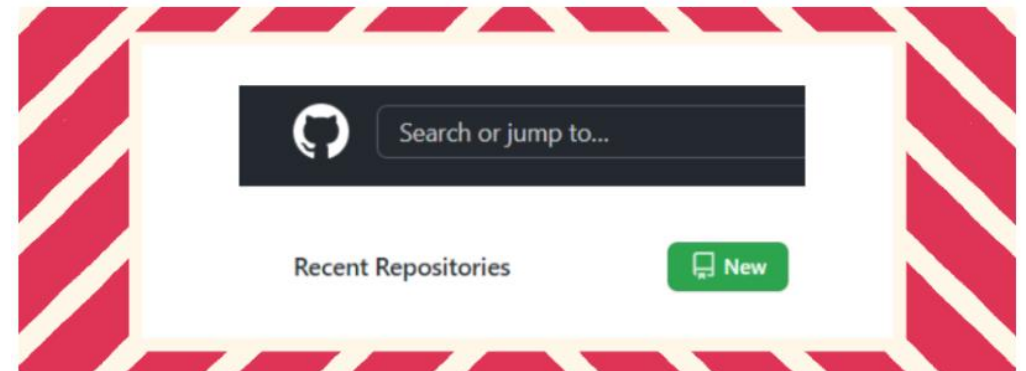


Github

INSTALACIÓN

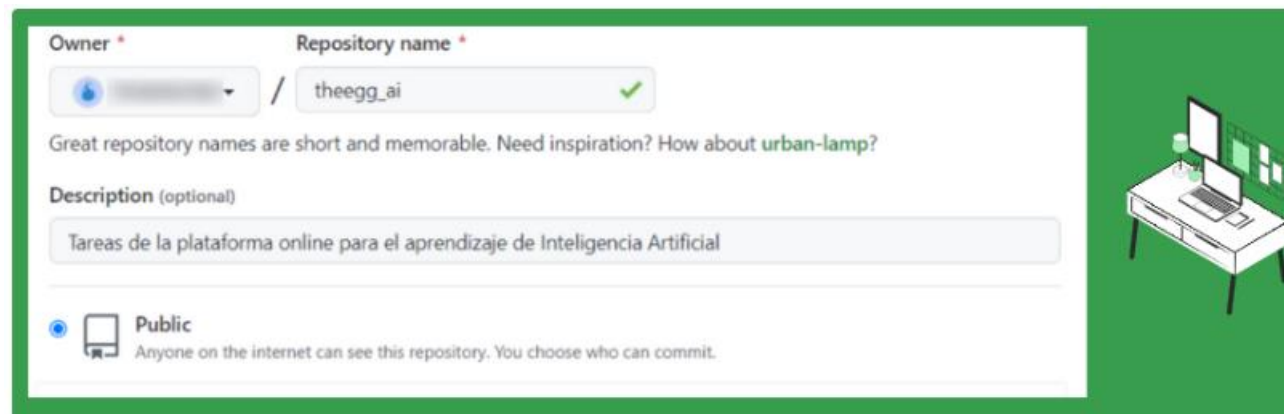
Instalación

- 1.- Debes abrirte una cuenta en [GitHub](#) registrándote en la plataforma.
- 2.- Una vez registrado debes iniciar sesión. Hasta aquí igual que en cualquier otra plataforma.
- 3.- Para crear la carpeta **theegg_ai** debes pulsar en el botón superior izquierdo donde pone *new*.



Instalación

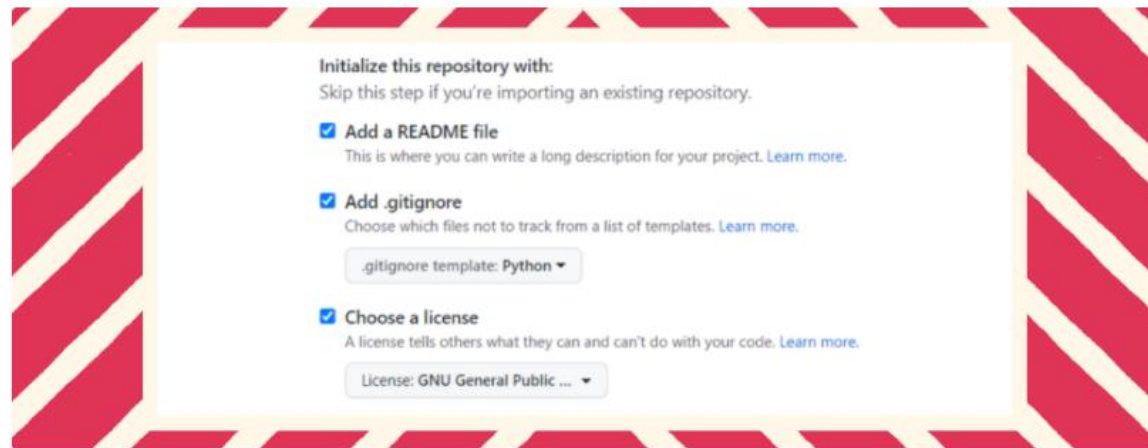
- 4.- Se te abrirán unos campos que debes rellenar: el *nombre del repositorio* y la *descripción*. A la hora de poner el nombre del repositorio debes ser muy cauteloso y no confundirte. En el caso de The Egg el nombre exacto es: **theegg_ai**
- 5.- Debes poner una breve descripción de los contenidos que vas a subir a dicha carpeta y dejar el repositorio como público (de lo contrario The Egg no va a poder verificar si tus trabajos han sido subidos).



The screenshot shows the GitHub repository creation interface. The 'Owner' field is set to a user profile. The 'Repository name' field contains 'theegg_ai' with a green checkmark. Below this, a hint suggests 'urban-lamp?'. The 'Description (optional)' field contains 'Tareas de la plataforma online para el aprendizaje de Inteligencia Artificial'. The 'Public' radio button is selected, with a note stating 'Anyone on the internet can see this repository. You choose who can commit.' To the right of the form is an illustration of a desk with a laptop and a monitor.

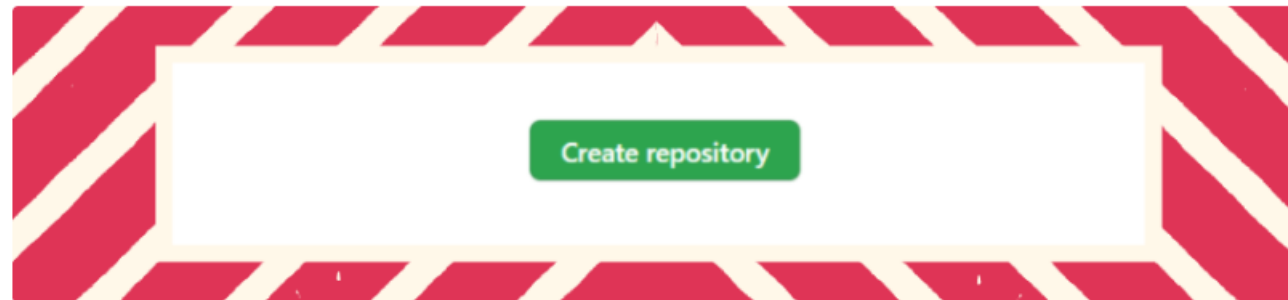
Instalación: recomendaciones

- a.- Activar Add a README file: es un fichero donde poder explicar modos de uso, notas importantes, etcétera sobre el contenido de la repo. Se puede editar.
- b.- Vincular el Add .gitignore a Python: para indicar a que tipo de categoría pertenece el repositorio. Se encapsulará en la agrupación a la que se refiera.
- c.- Seleccionar en Choose a license a una licencia GNU. Es una licencia ampliamente usada en el mundo del software libre y código abierto.



Contexto: modelos de negocio

Una vez rellenados los campos pulsa el botón *Create repository* y ya estarás listo para trabajar:



TRABAJAR CON GITHUB

Trabajar con Github

Tanto Git como GitHub ofrecen a los programadores una valiosa funcionalidad de control de versiones para que puedas construir proyectos de IA sin tener miedo a estropearlo todo. GitHub simplemente lleva las cosas un poco más lejos que Git, ofreciendo más funcionalidad y recursos.



Trabajar con Github

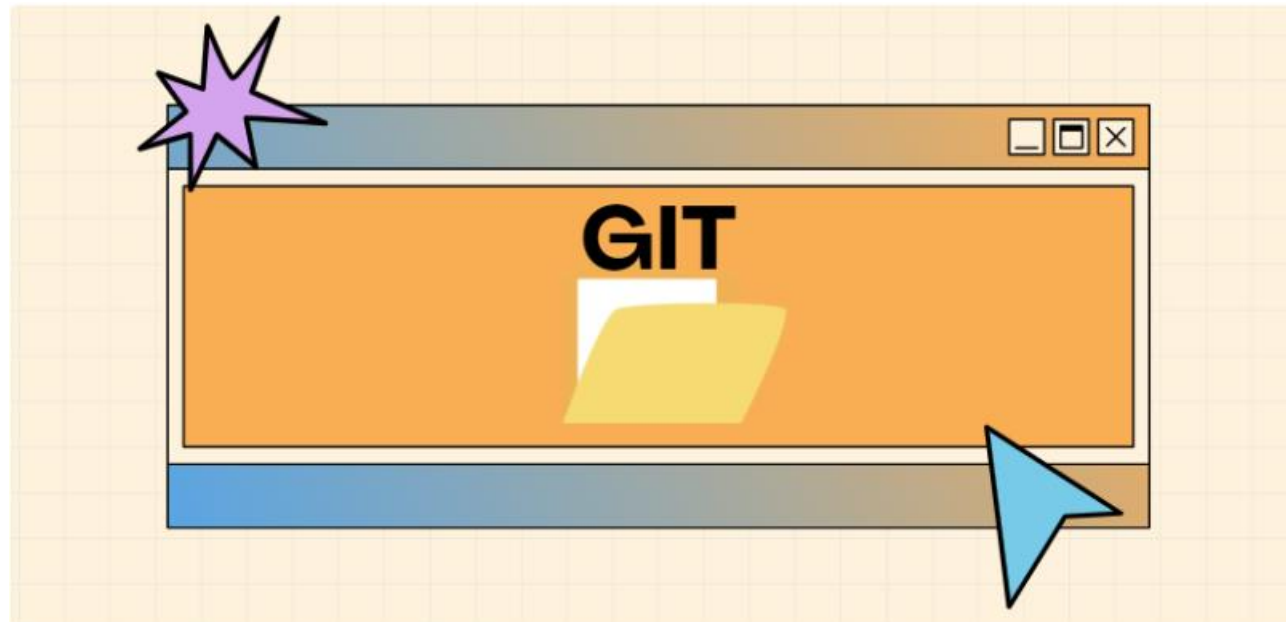
Necesitamos instalar Git en nuestra computadora para que GitHub pueda gestionar repositorios Git.

En Windows:

- 1.- Debes ir a <http://git-scm.com/download/win> y descargarte el paquete que necesitas.
- 2.- Una vez descargado el instalador lo ejecutas y siguiente - siguiente - ...

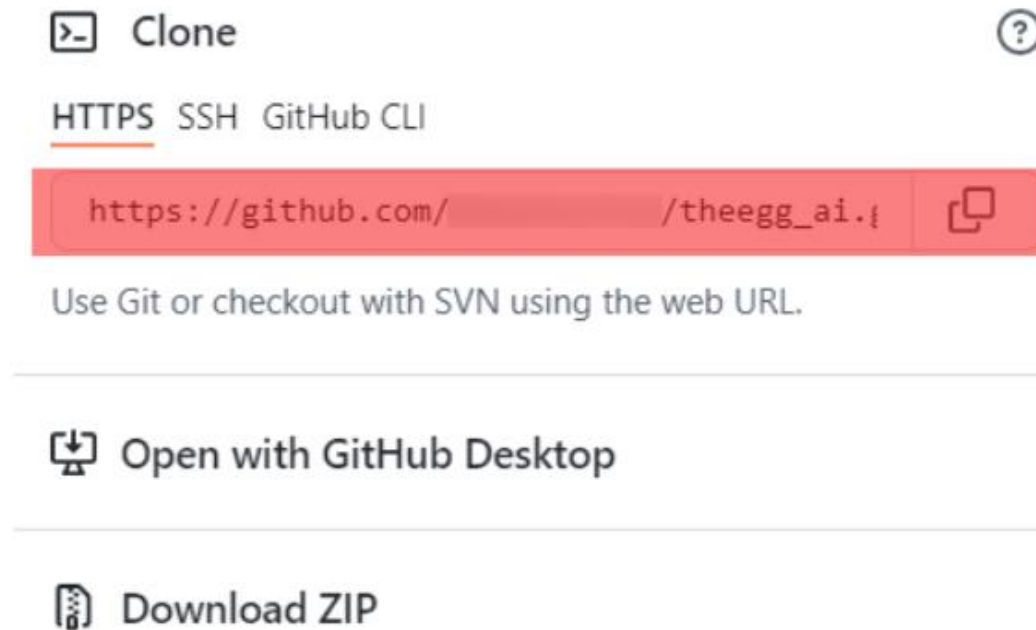
Trabajar con Github: pasos

1.- Abre una carpeta llamada GIT en tu escritorio (o donde quieras)



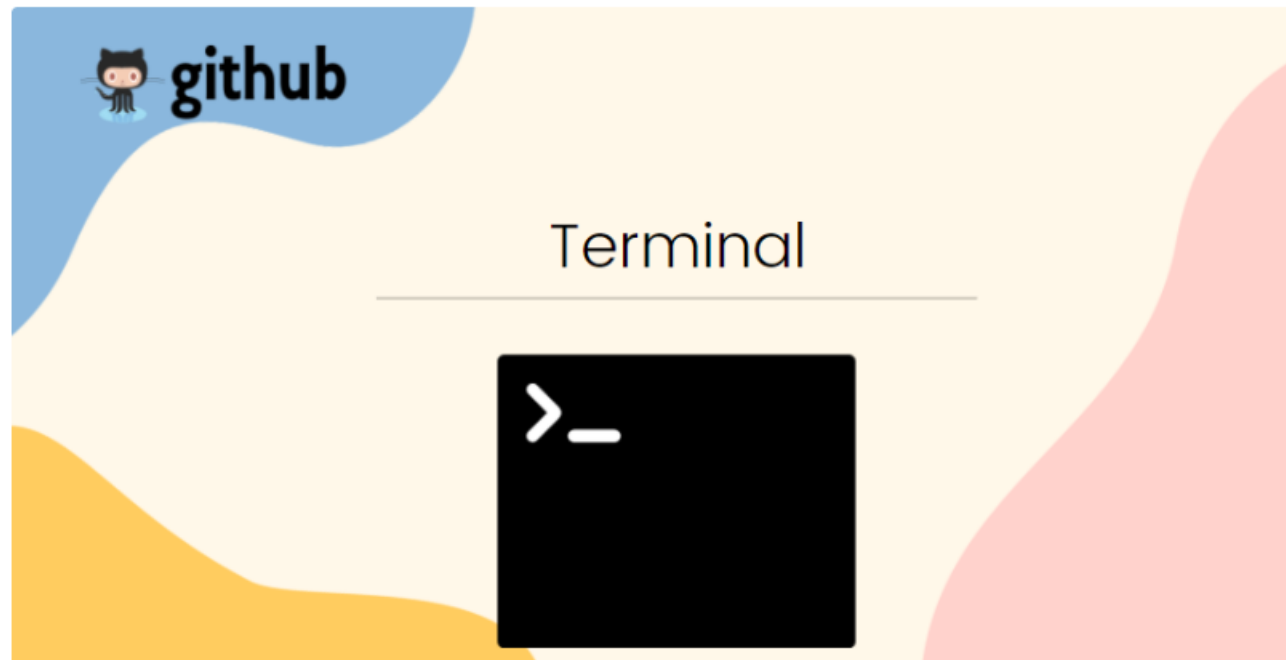
Trabajar con Github

2.- Entra a tu repo en Github creado para TheEgg y copia el enlace coloreado en rojo:



Trabajar con Github

3.- Entra en la carpeta que acabas de crear, pulsa el botón derecho del ratón y selecciona "*Abrir el Terminal Windows en vista previa*".



Trabajar con Github: clone

4.- Una vez en el terminal escribe lo siguiente: git clone + la dirección que has copiado

```
git clone https://github.com/tu_nombre_de_usuario/theegg_ai.git
```

Trabajar con Github: clone

5.- Verás algo parecido a esto:

```
PS C:\Users\Usuario\Desktop\GIT> git clone https://github.com/tu_nombre_de_usuario/th
Cloning into 'theegg_ai'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), 13.52 KiB | 3.38 MiB/s, done.
```

6.- Verás que se ha hecho una copia exacta del repositorio *theegg_ai* en tu carpeta.

7.- Felicidades. Ya estás operando con Github.

Trabajar con Github: status

- 1.- Dentro de la copia *theegg_ai* que se ha escrito en tu directorio crea un fichero *hola.txt* (con el block de notas) y escribe dentro del fichero de texto lo siguiente: *Hola Mundo*.
- 2.- Ubícate dentro de *theegg_ai* y abre el terminal. ¡¡¡Pero cuidado!!!
El terminal tiene que estar ubicado en el directorio en el que estás trabajando, de lo contrario se volverá loco.

```
C:\Users\Usuario\Desktop\GIT\theegg_ai>
```

Trabajar con Github: status

En caso de abrir el terminal desde otra ubicación te tendrás que desplazar hasta la carpeta *theegg_ai* desde el propio terminal mediante la instrucción *cd* (este comando sirve tanto para Windows como para Linux). Por ejemplo, imagínate que abres el terminal desde el escritorio y necesitas desplazarte a la carpeta *GIT/theegg_ai* que está en el mismo escritorio:

```
PS C:\Users\Usuario\Desktop> cd GIT/theegg_ai  
PS C:\Users\Usuario\Desktop\GIT\theegg_ai>
```


Trabajar con Github: status

3.- Una vez que estés en la carpeta escribe *git status*:

```
PS C:\Users\Usuario\Desktop\GIT\theegg_ai> git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Usuario\Desktop\GIT\theegg_ai>
```

Esta instrucción nos informa de los cambios que han ocurrido en la carpeta (nuevos ficheros o cambios dentro de un mismo fichero). En este caso existe un nuevo fichero *hola.txt*

Trabajar con Github:add

- 1.- Escribe `git add hola.txt`. Parece que no hace nada pero realmente ha añadido `hola.txt` al *staging area* (la zona de puesta en escena). También podríamos escribir `git add .` y subiría todos los archivos que han sufrido algún cambio.
- 2.- Si cambiásemos el contenido de `hola.txt` y escribiésemos `git diff` el sistema nos informaría de las diferencias del archivo local con respecto al archivo que está en el *staging area*.

Trabajar con Github: commit

- 1.- Para encapsular todo en la repo local antes de subirlo a la nube debes escribir: *git commit*. Previamente te tendrás que identificar metiendo *tu correo* y *tu Nick* de Github
- 2.- Si encima deseas añadir una explicación a la nueva versión que pretendes subir puedes escribir lo siguiente: *git commit -m "explicación que consideres oportuna"*
Te quedaría algo parecido a esto:

```
PS C:\Users\Usuario\Desktop\GIT\theegg_ai> git config --global user.email "tucorreo@
PS C:\Users\Usuario\Desktop\GIT\theegg_ai> git config --global user.name "tuNick"
PS C:\Users\Usuario\Desktop\GIT\theegg_ai> git commit -m "Nueva versión con fichero h
[main 160eba6] Nueva versión con fichero hola.txt
 1 file changed, 1 insertion(+)
 create mode 100644 hola.txt
PS C:\Users\Usuario\Desktop\GIT\theegg_ai>
```





Trabajar con Github: push

- 1.- El último paso para subir el contenido actualizado a nuestro GitHub es escribir: *git push*
- 2.- Evidentemente el sistema te pedirá *tu Nick* y *tu contraseña* de Github antes de hacer el *push*. Finalmente quedará algo parecido a esto:

```
PS C:\Users\Usuario\Desktop\GIT\theegg_ai> git push
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 301 bytes | 301.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/tu_Nick/theegg_ai.git
   96e3cbd..160eba6  main -> main
PS C:\Users\Usuario\Desktop\GIT\theegg_ai>
```

Trabajar con Github: push

3.- Puedes ir a Github a tu repo *theegg_ai* y observarás que se han subido tus cambios.

 .gitignore	Initial commit	7 days ago
 LICENSE	Initial commit	7 days ago
 README.md	Initial commit	7 days ago
 hola.txt	Nueva versión con fichero hola.txt	22 minutes ago

Trabajar con Github

LO QUE HEMOS VISTO

comandos	Tu directorio	Puesta en escena	Repositorio local	Repositorio remoto
git add .				
git commit -m "Comentario descriptivo"				
git push				
git clone direccion.git				