**REPOSITORIO GITHUB**

# ***¿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

## **¿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.

## **¿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.

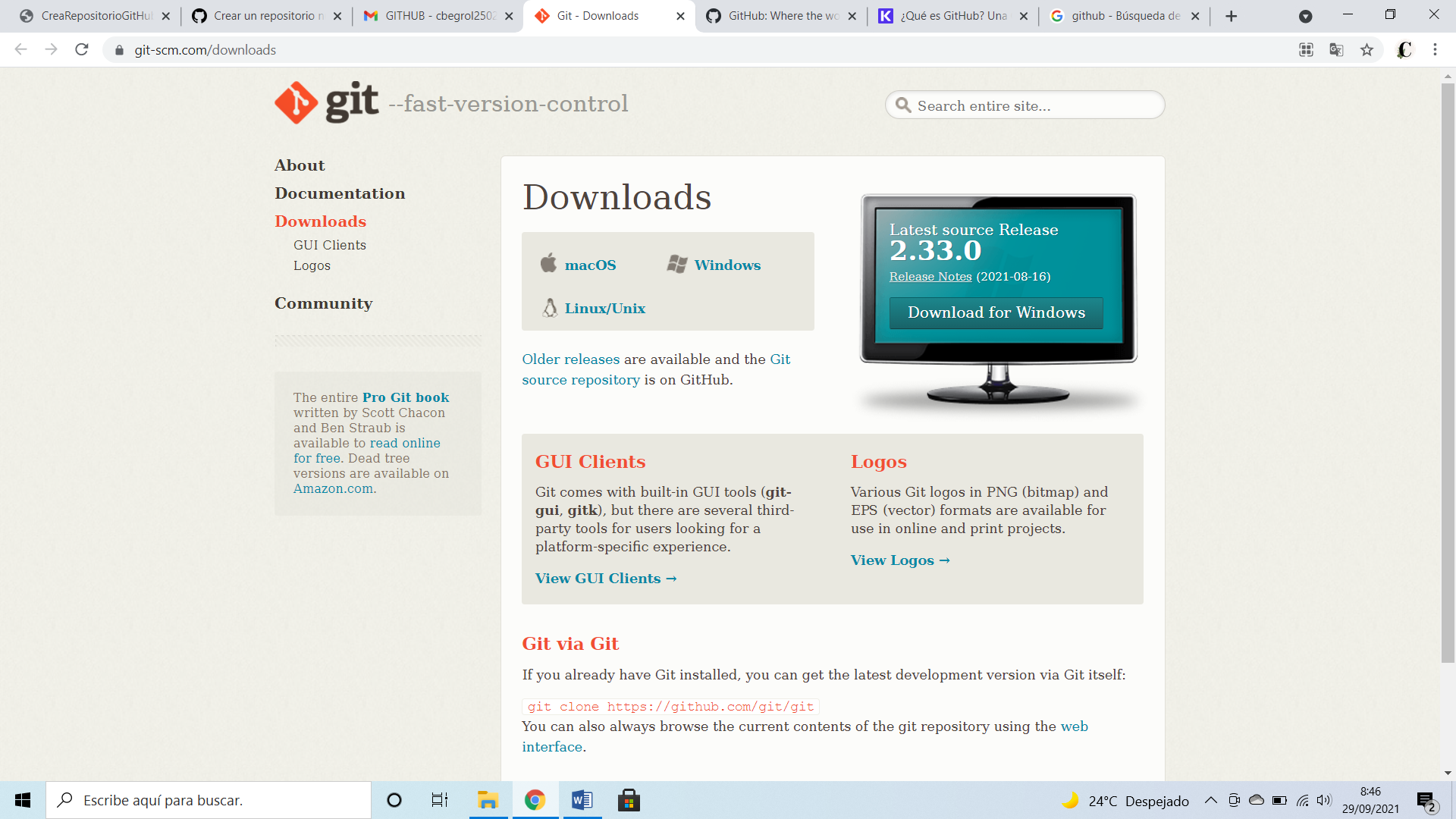
# ***Comenzar con GitHub.***

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

## **Descargar GitHub**

Podemos descargar GitHub fácilmente mediante el enlace:

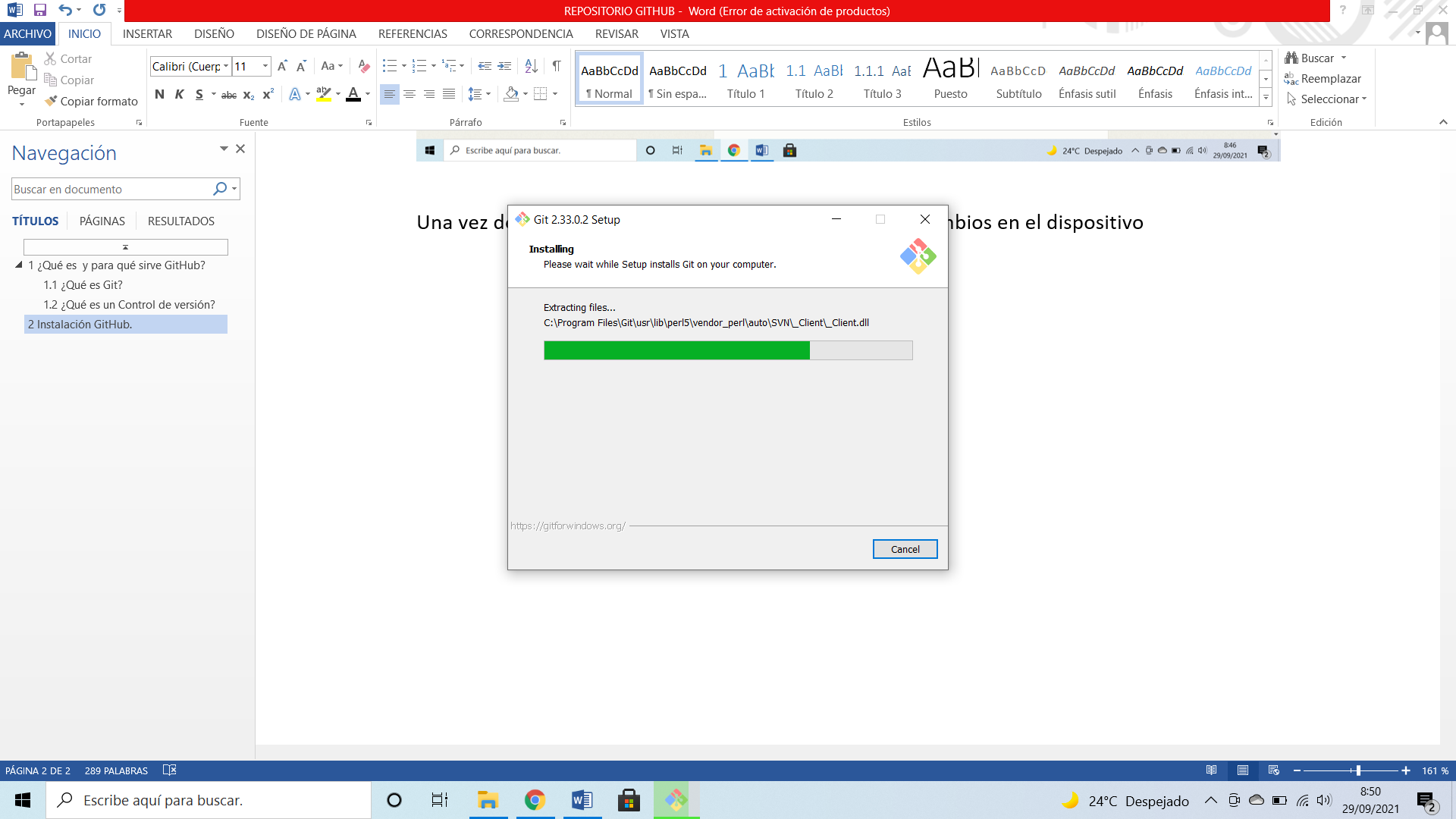
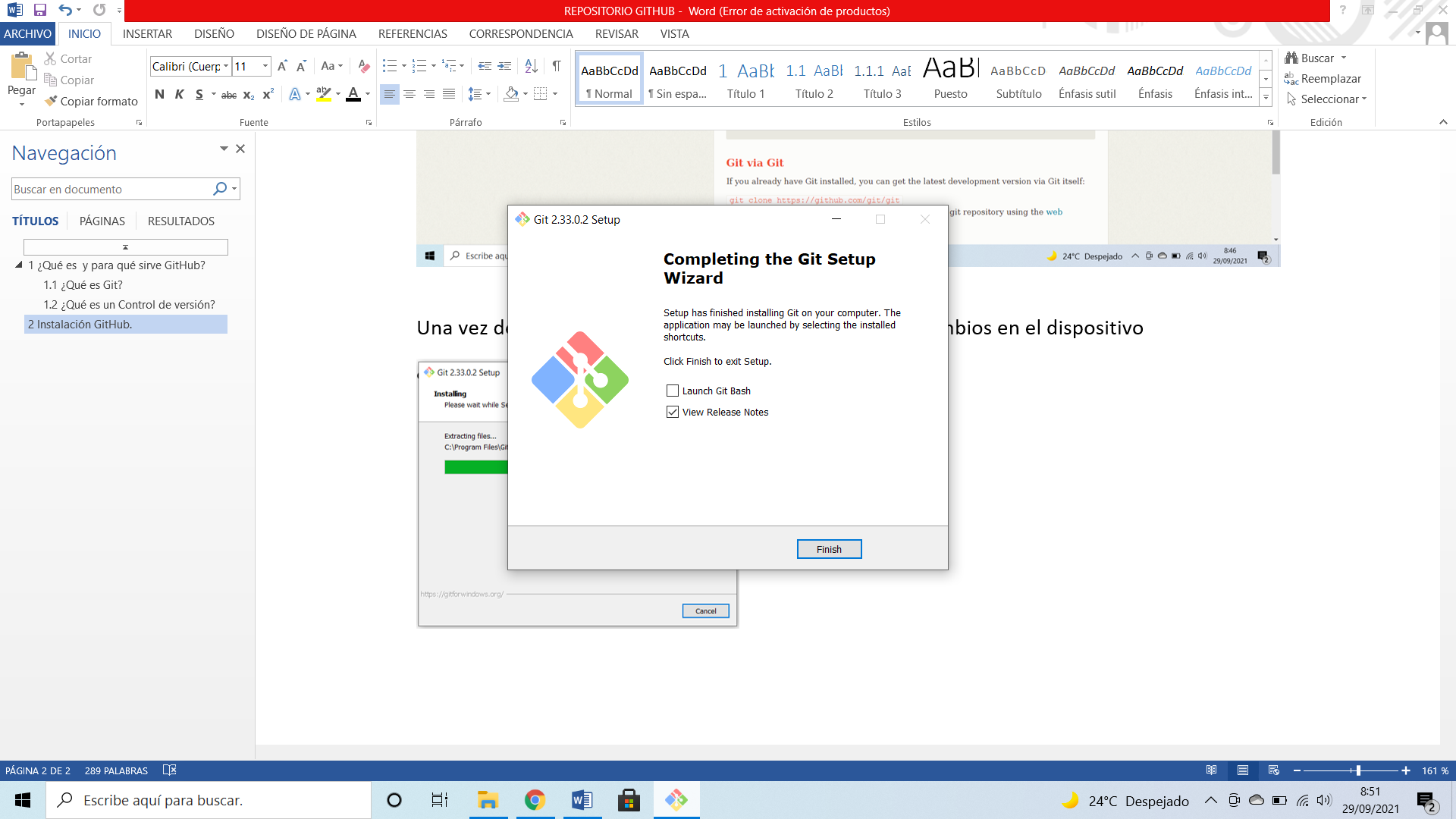
<https://git-scm.com/downloads>

En este caso, lo descargaremos para Windows.

## **Instalación**

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

Se completa rapidamente y finalizamos la instalación.

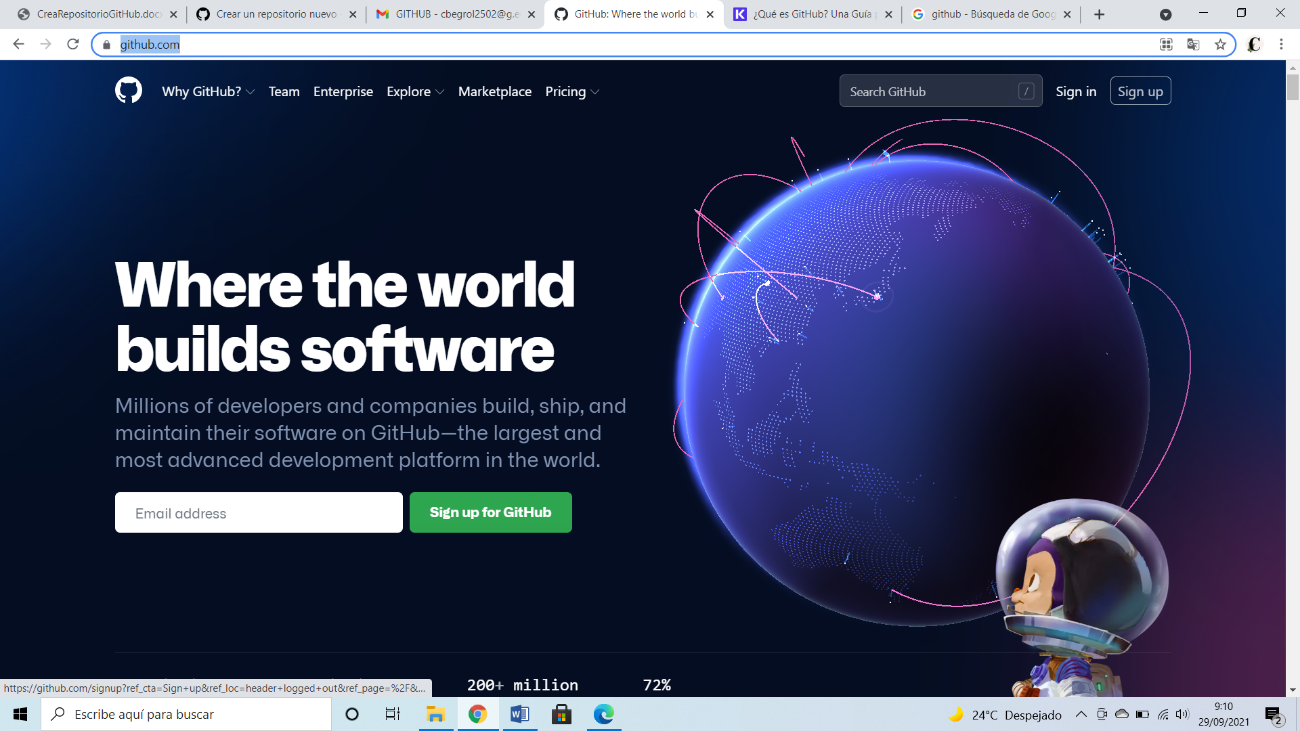


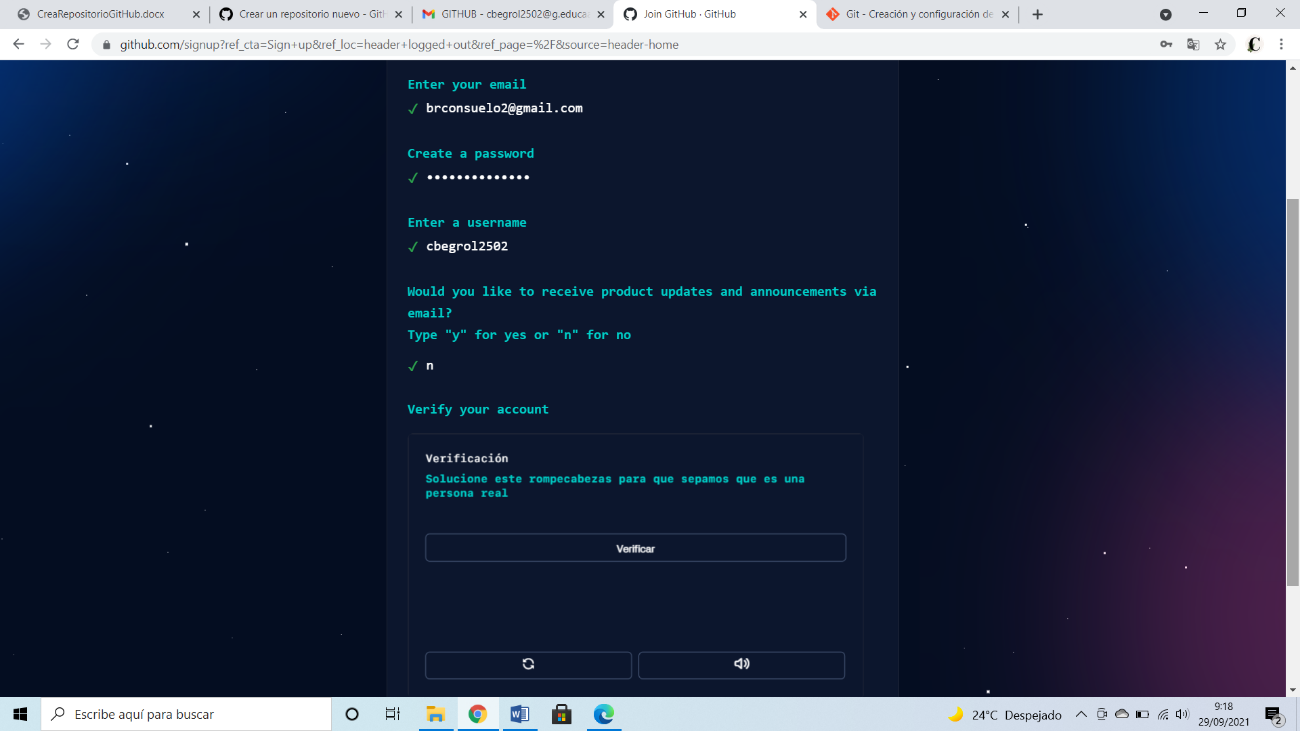
## **Creación y configuración de la cuenta**

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.

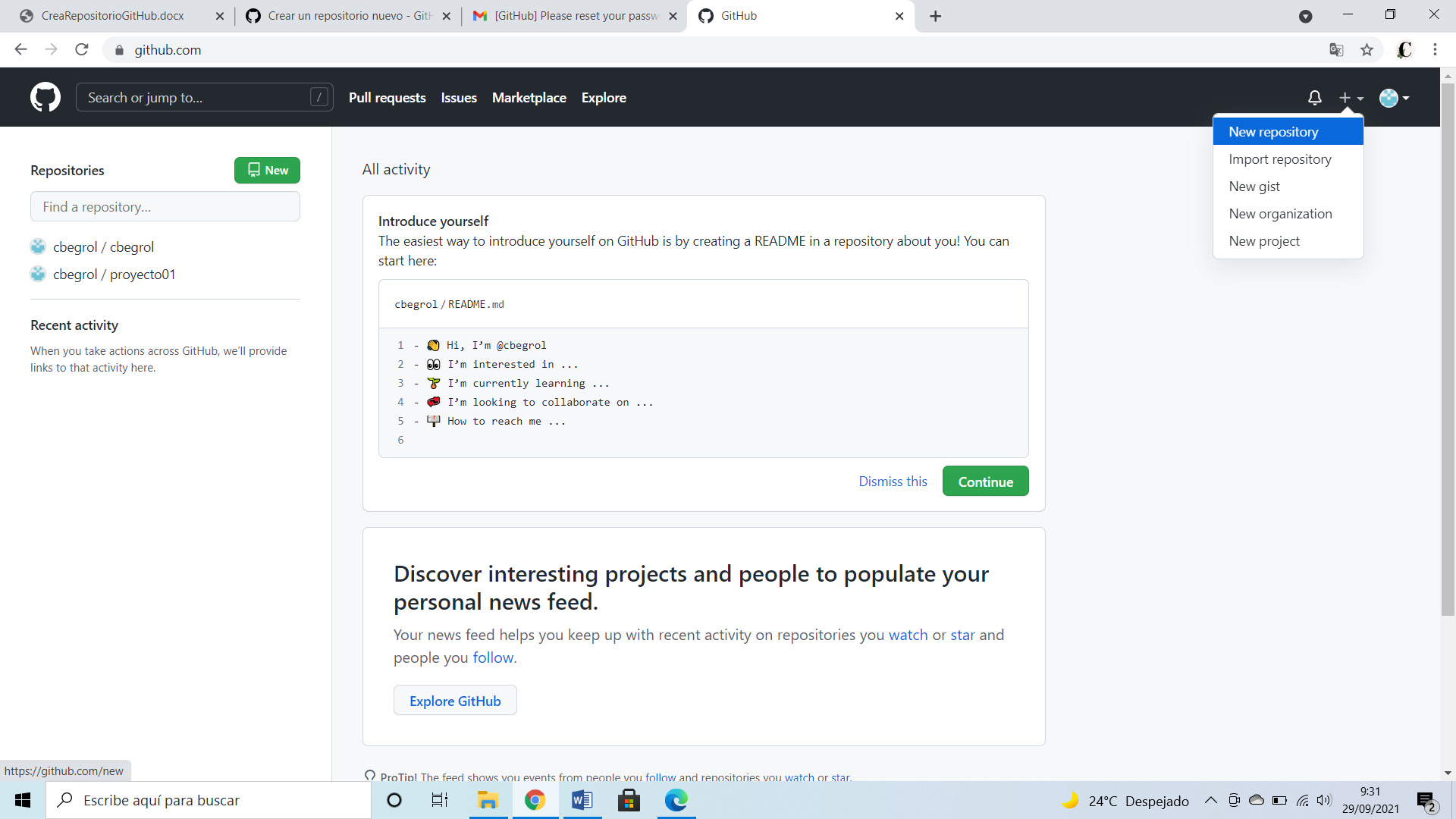
Solucionamos un rompecabezas y así, verificamos la cuenta.

Ahora, ya tenemos cuenta de GitHub.

# ***Creación de un nuevo repositorio***

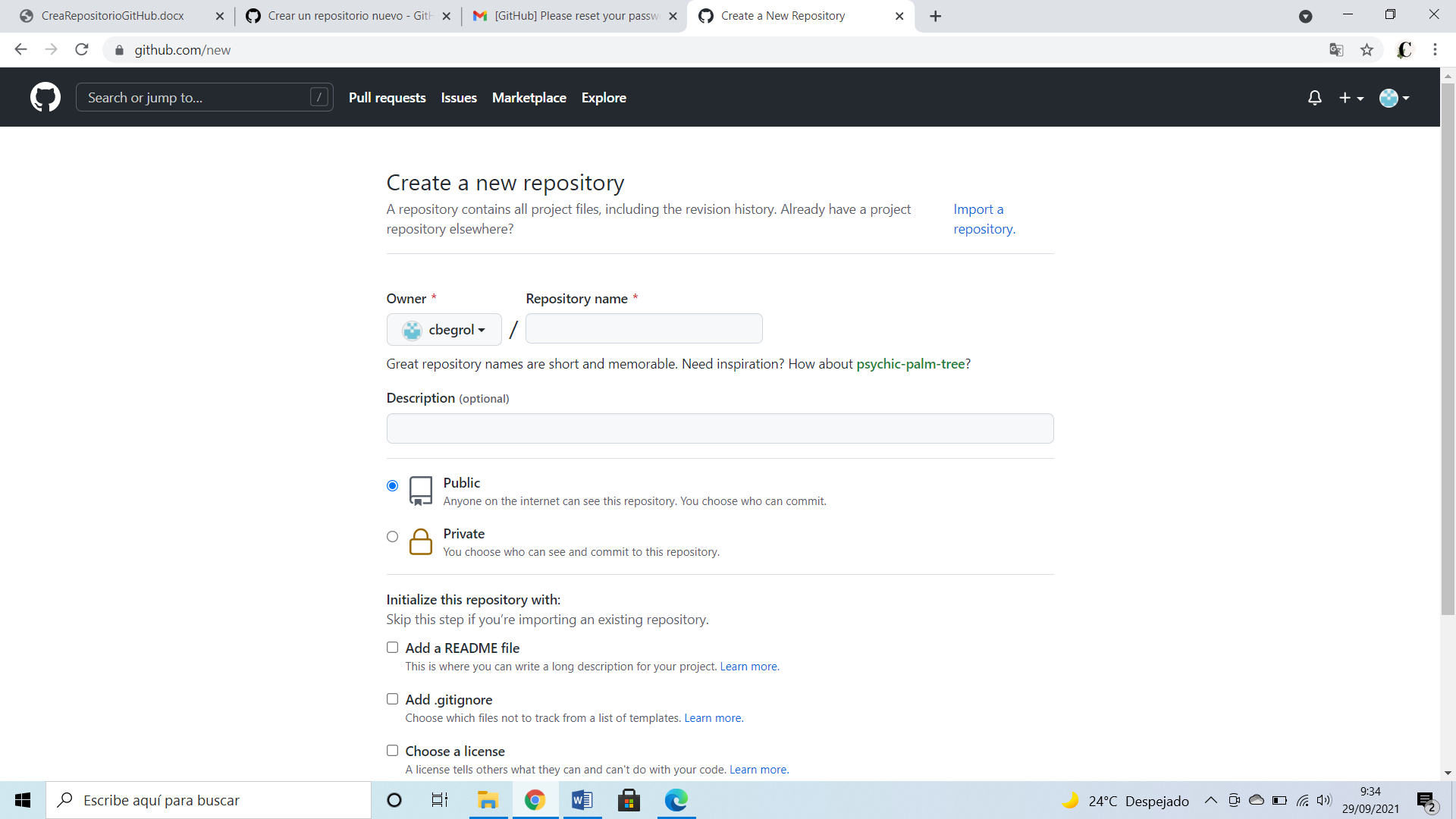
## **Paso 1:**

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



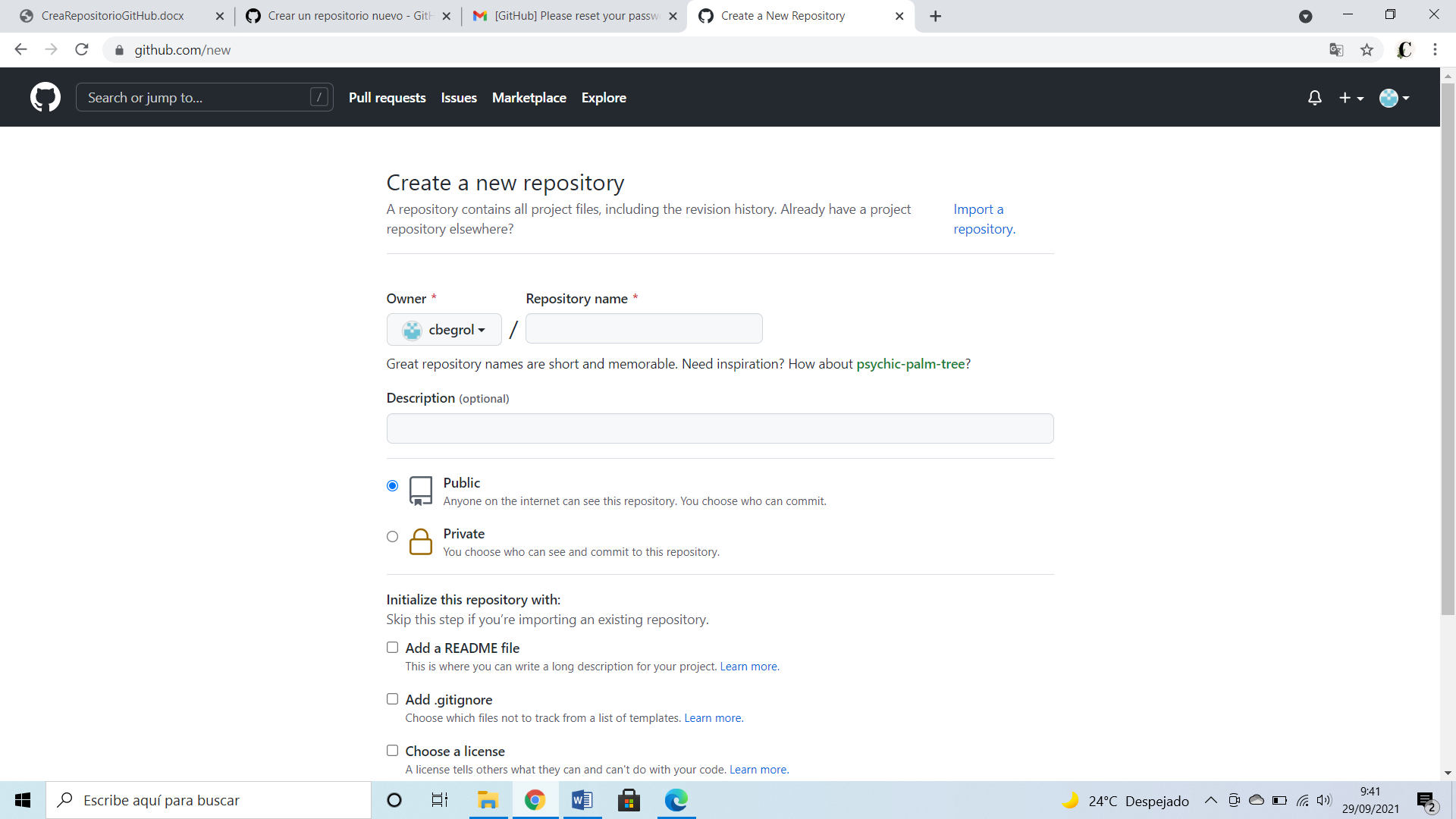
## **Paso 2:**

## 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.



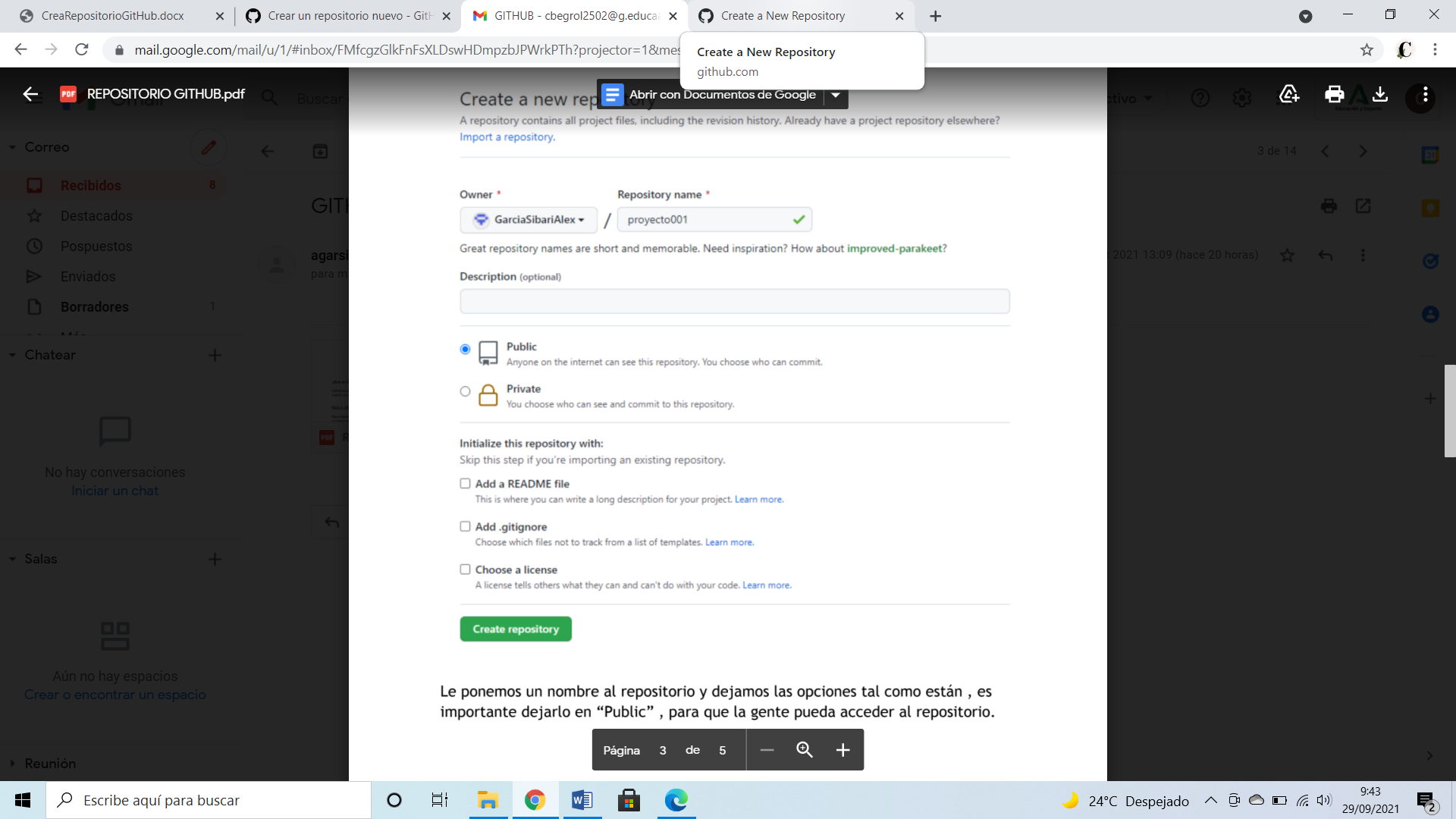
## **Paso 3:**

## Elige la visibilidad del repositorio.



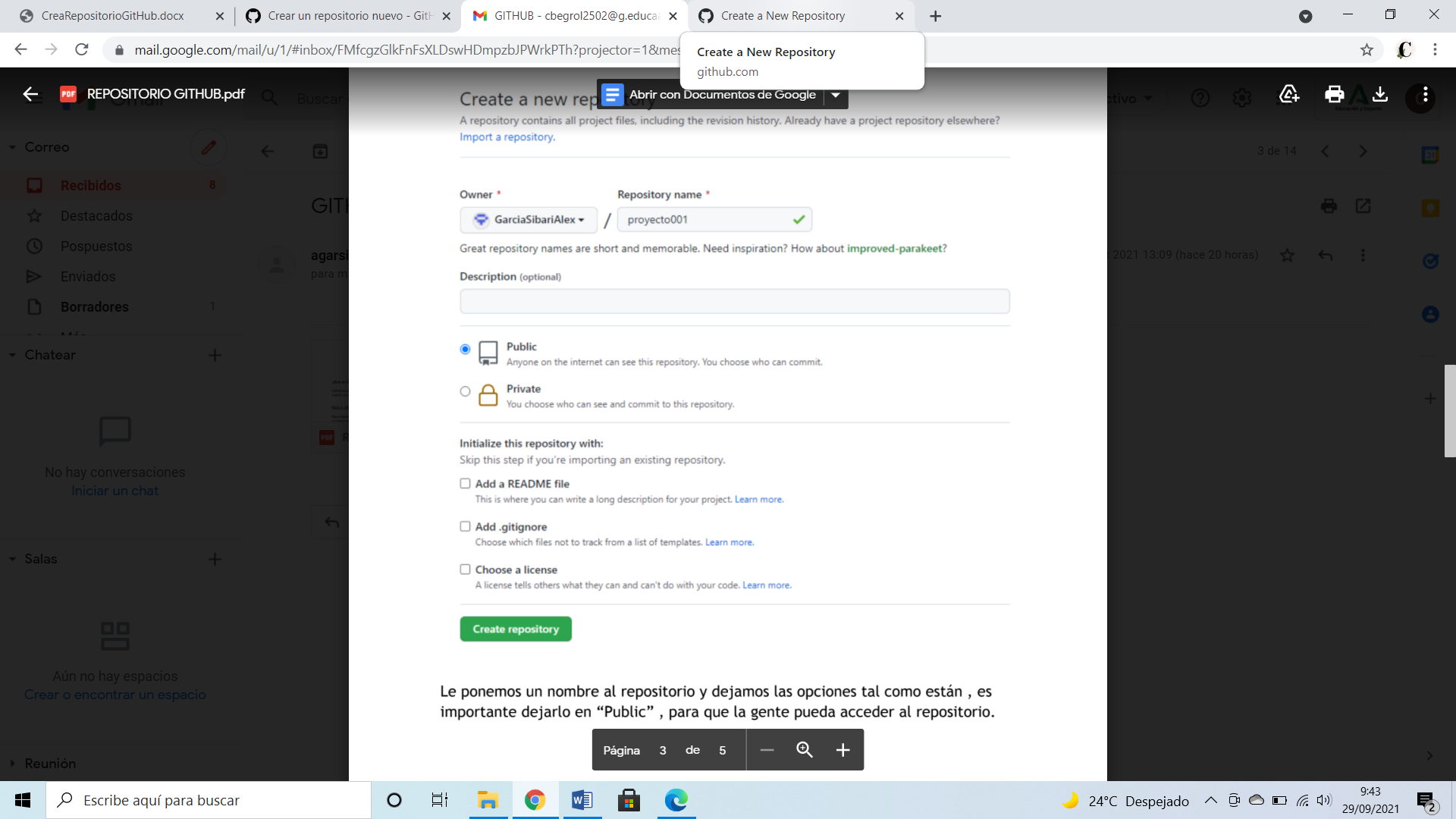
## **Paso 4:**

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

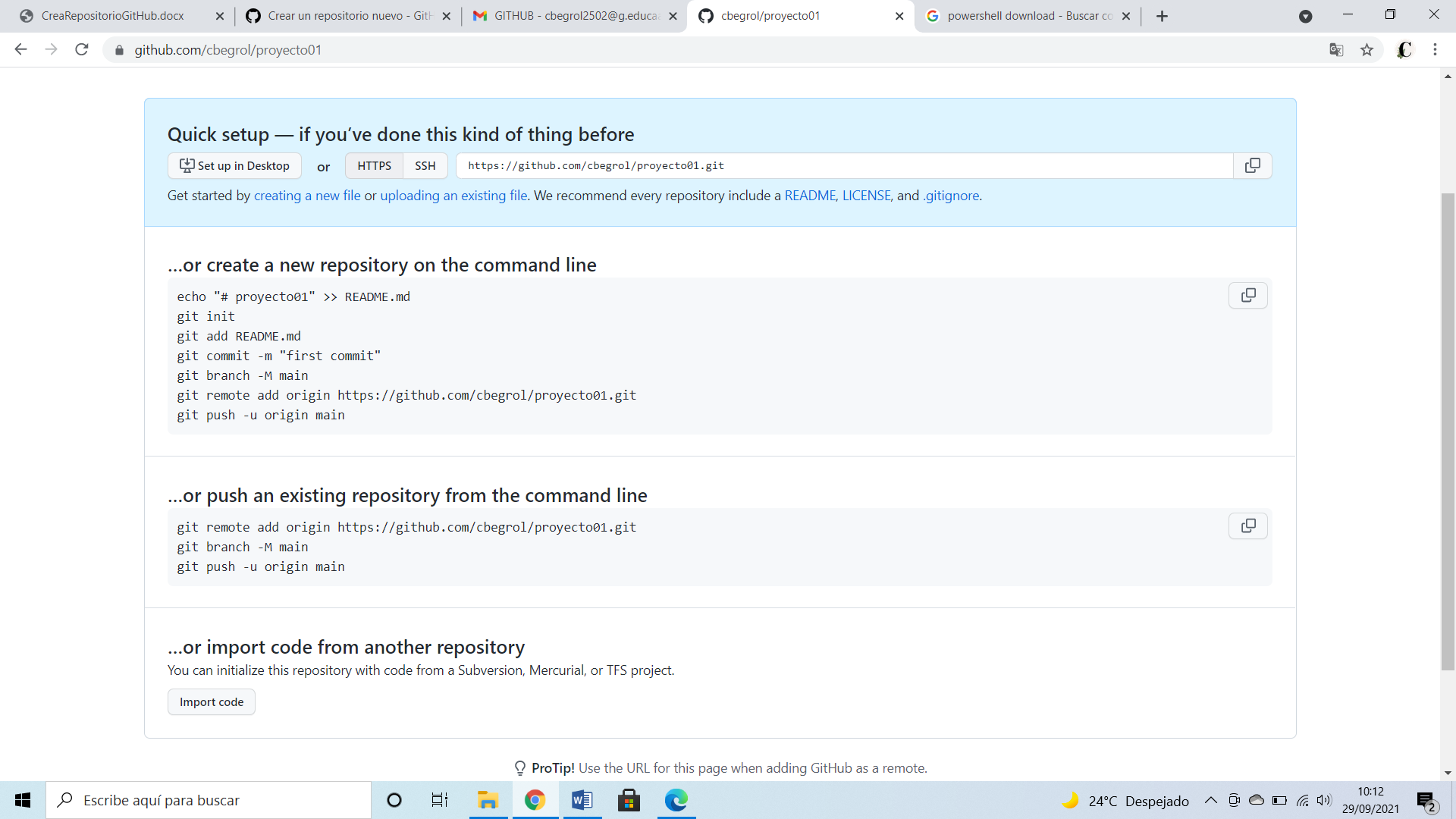


## **Paso 5:**

## 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.



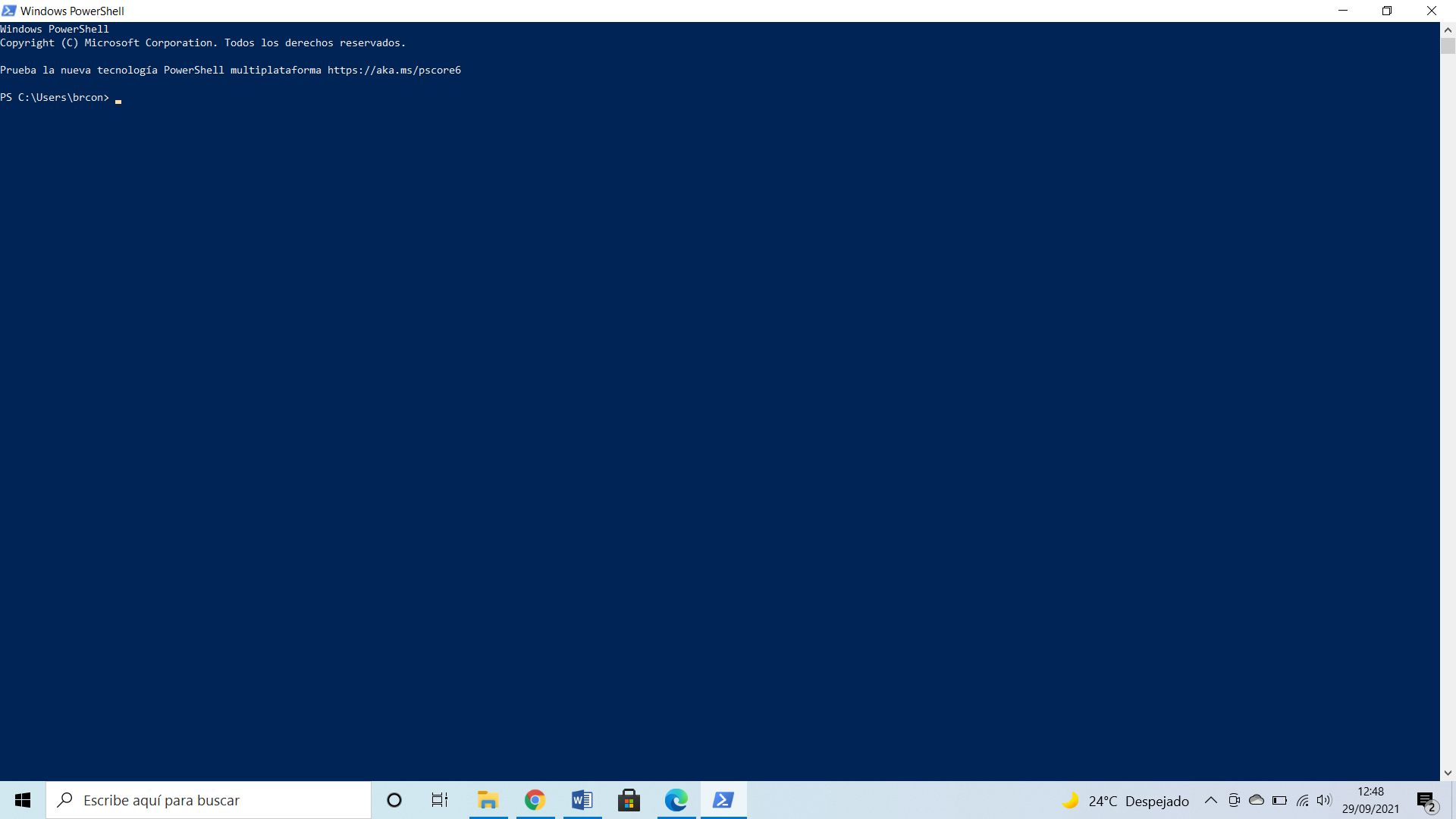
# **Windows PowerShell**

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

## **Abrir programa**

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

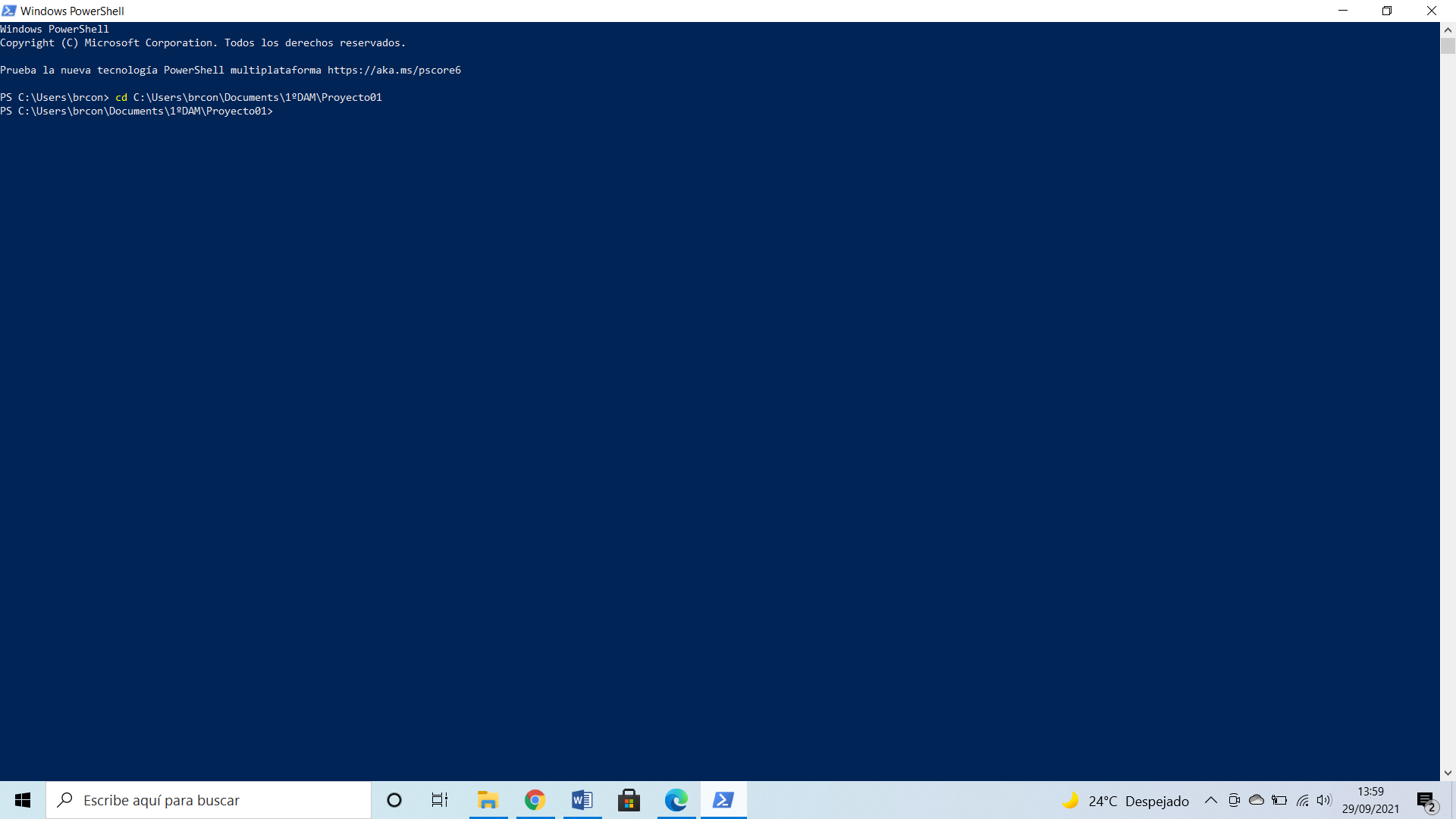
Una vez lo abrimos, nos aparece esto:



## **Comando cd**

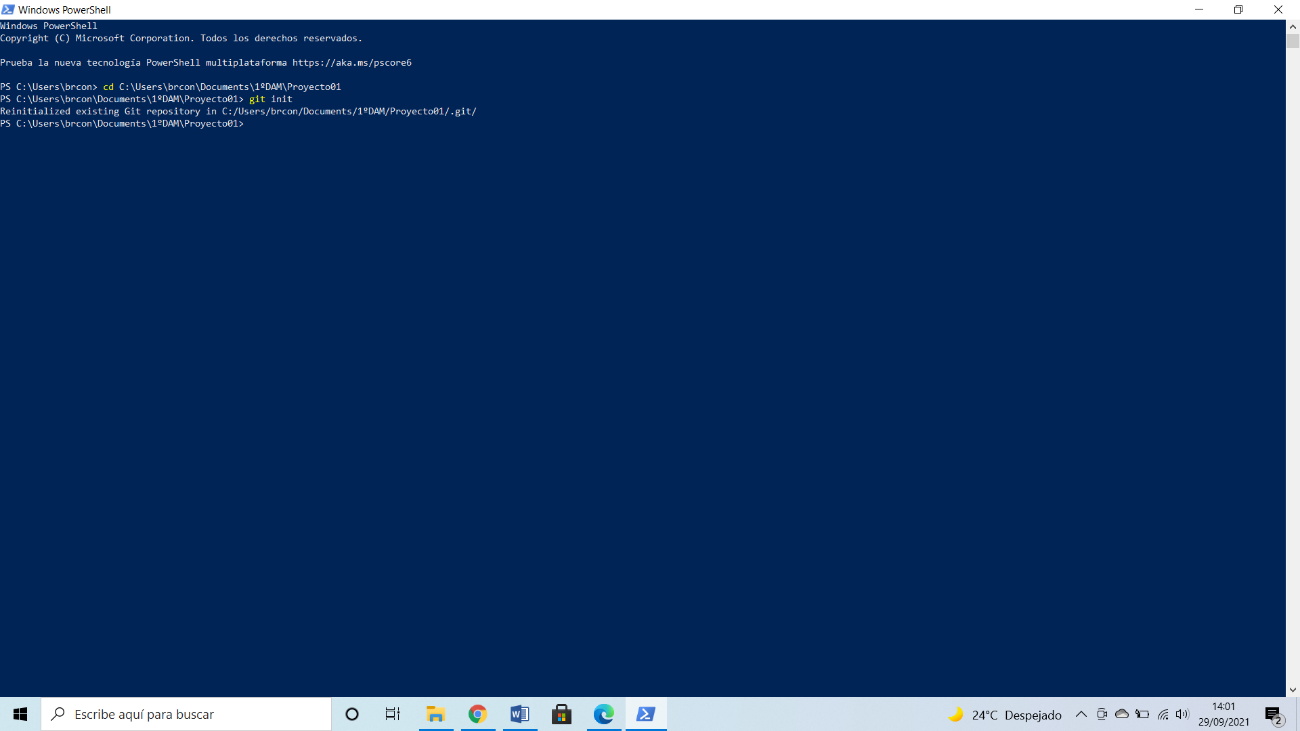
Comenzamos con el comando cd, uno de los comandos más esenciales de la consola de Windows. Sirve para cambiar de directorio, utilizando la fórmula \*cd < RutaDirectorio >\* para ir al directorio o carpeta concreta que le digas, o \*cd..\* (con dos puntos) para salir de una carpeta e ir al nivel superior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



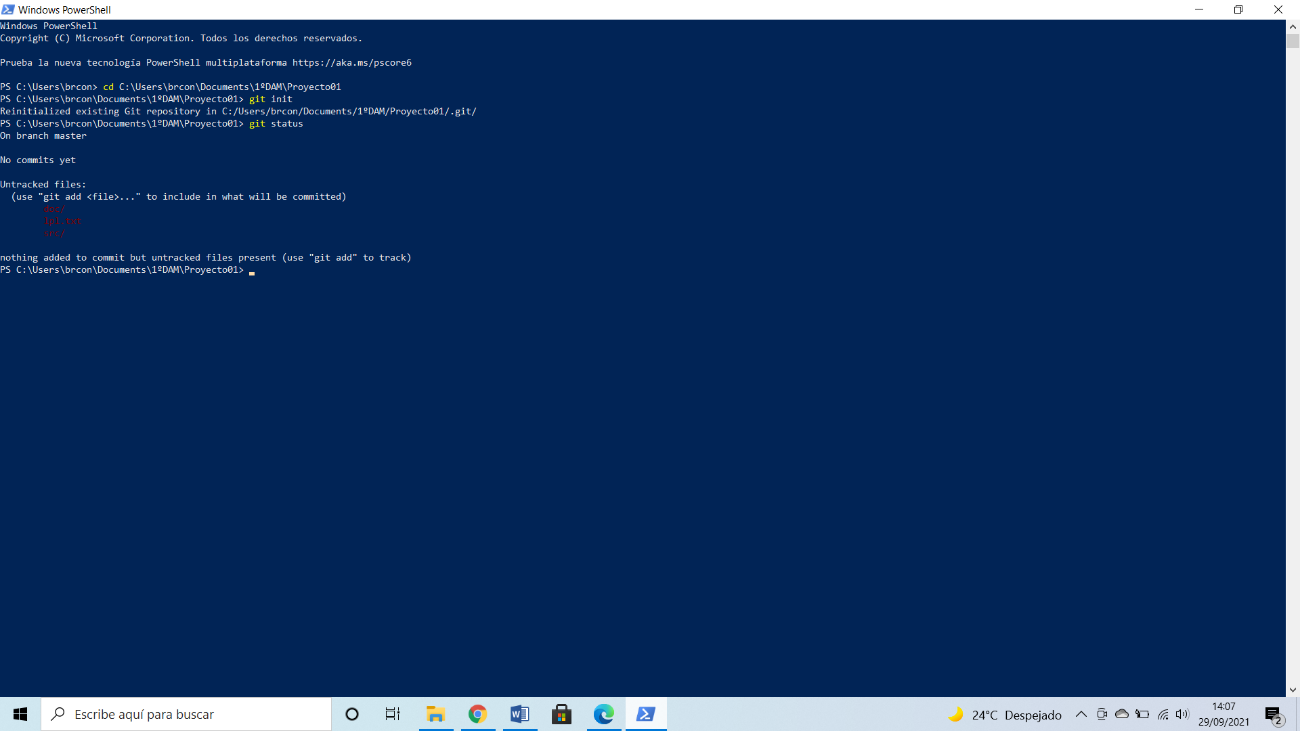
## **Comando git init**

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.



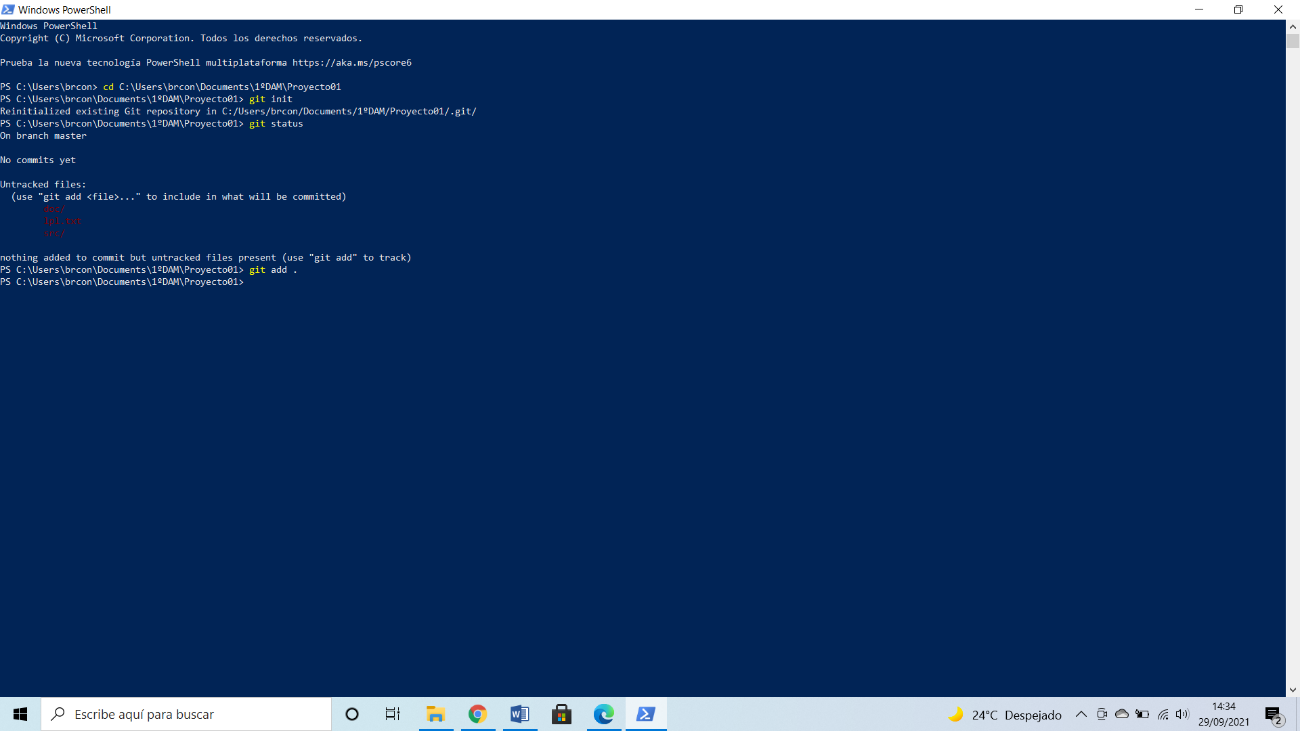
## **Comando git status**

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.



## **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.



## **Comando git commit**

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.