

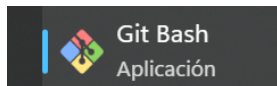
GitHub

1.- Crear repositorio remoto VACÍO en (GitHub)

...

2.- Convertir una carpeta local en repositorio y conectarlo con un repositorio remoto (ya creado)

Abrir la aplicación



Situarnos en la carpeta que queremos convertir en repositorio y:

```
echo "# prb" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
```

La primera línea solo es estrictamente necesaria si la carpeta local no contiene archivos.

Si la carpeta local ya contiene archivos, cambiaríamos la tercera línea por “git add .” para que añada todos los archivos de la misma.

Elegir el protocolo a utilizar (https o ssh) con el que más cómodo te sientas

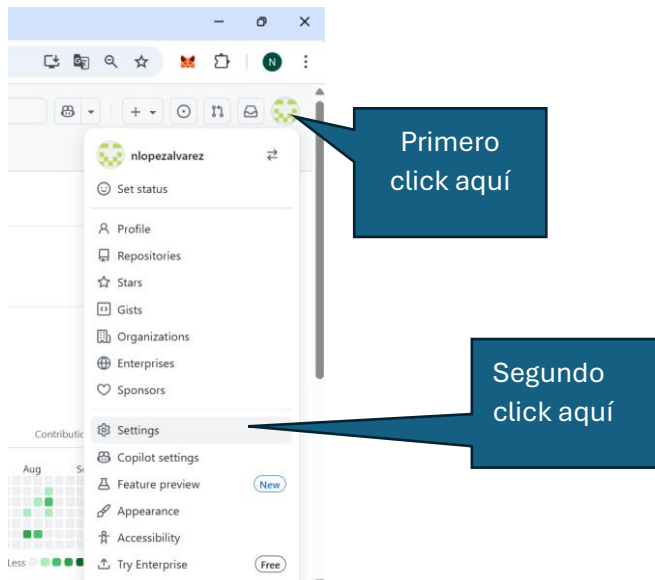
Con https

```
git remote add origin https://github.com/nlopezalvarez/prb.git
git push -u origin main
```

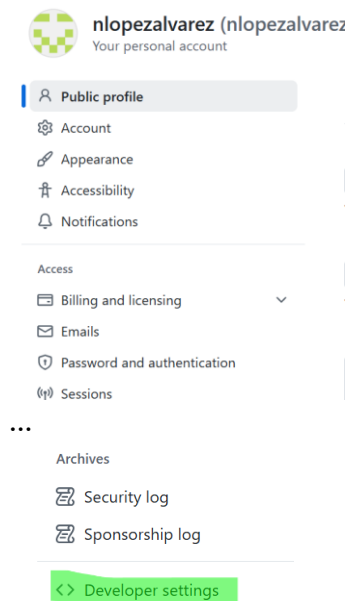
Si pide nombre de usuario y contraseña, la contraseña de nuestra cuenta de GitHub no será válida y tendremos que crear un token, que usaremos como contraseña.

Si no disponemos de un token, se crea de la siguiente forma:

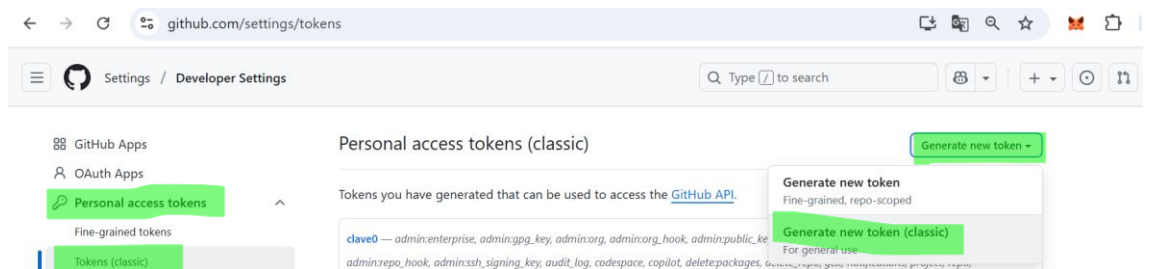
Entramos en nuestra cuenta de GitHub y luego en Settings, que se encuentra como se muestra a continuación:



En el panel izquierdo, en la parte inferior, seleccionamos “Developer settings”:



A continuación, seleccionamos “Personal Access tokens” y en el desplegable “Tokens (classic)”. Después pulsamos el botón “Generate new token” y en el desplegable “Generate new token (classic)”:



Ahora escribimos (en Note) el nombre que queramos y en la sección “Select scopes” marcamos todas las casillas

tokens/new

Q Type to search

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

nombreQueQueramosParaElToken

What's this token for?

Expiration

30 days (Oct 31, 2025)

The token will expire on the selected date

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
...	
<input type="checkbox"/> codespace:secrets	Ability to create, read, update, and delete codespace secrets
<input checked="" type="checkbox"/> copilot	Full control of GitHub Copilot settings and seat assignments
<input type="checkbox"/> manage_billing:copilot	View and edit Copilot Business seat assignments
<input checked="" type="checkbox"/> write:network_configurations	Write org hosted compute network configurations
<input type="checkbox"/> read:network_configurations	Read org hosted compute network configurations
<input checked="" type="checkbox"/> project	Full control of projects
<input type="checkbox"/> read:project	Read access of projects
<input checked="" type="checkbox"/> admin:pgp_key	Full control of public user GPG keys
<input type="checkbox"/> write:pgp_key	Write public user GPG keys
<input type="checkbox"/> read:pgp_key	Read public user GPG keys
<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

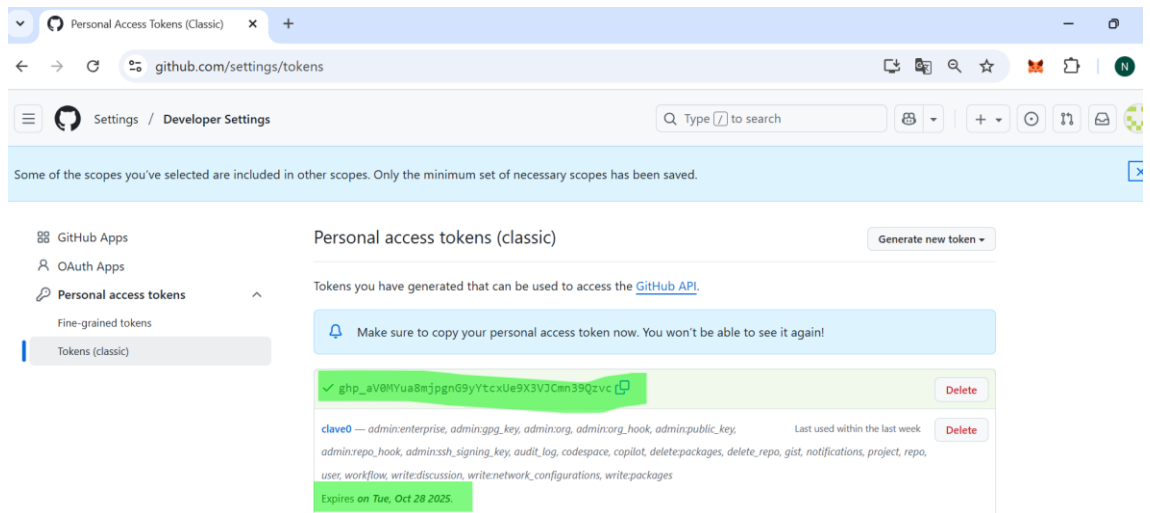
Generate token Cancel

Pulsamos el botón “Generate token”.

A continuación, se muestra nuestro token (para utilizarlo como contraseña).

Copiarlo y guardarlo en un archivo para poder usarlo como contraseña cuando se nos pida.

Este token expira en la fecha indicada.



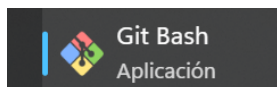
Este procedimiento se puede repetir tantas veces como se necesite. En caso de no disponer del token, se podría realizar de nuevo este proceso y volveríamos a disponer de él.

Ya podemos salir de GitHub y usar el protocolo https para manejar nuestros repositorios.

A partir de ahora, cuando git te pida nombre de usuario y contraseña, ésta será el token.

Con ssh

Abrir la aplicación

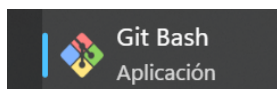


Situarnos en la carpeta del repositorio local y:

```
git remote add origin git@github.com:nlopezalvarez/prb.git
git push -u origin main
```

Si no tienes claves ssh, hay que crearlas. El procedimiento es el siguiente:

Abrir la aplicación



Y ejecutar el siguiente comando (no importa la carpeta en la que te encuentres):

```
$ ssh-keygen -t ed25519 -C tudireccion@de.correo
```

Te pedirá la ruta y nombre del fichero donde se almacenarán las claves:

```
Nuria López Álvarez@NuPC MINGW64 /c
$ ssh-keygen -t ed25519 -C tudireccion@de.correo
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Nuria López Álvarez/.ssh/id_ed25519):
```

Acepta el nombre de fichero y la ubicación por defecto pulsando Intro sin escribir nada.

A continuación, te pide una contraseña para proteger la clave privada:

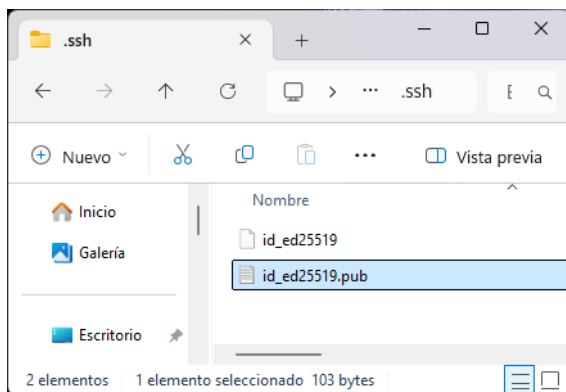
```
Nuria López Álvarez@NuPC MINGW64 /c
$ ssh-keygen -t ed25519 -C tudireccion@de.correo
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Nuria López Álvarez/.ssh/id_ed25519):
Enter passphrase for "/c/Users/Nuria López Álvarez/.ssh/id_ed25519" (empty for no passphrase):
```

Puedes pulsar Intro directamente y dejarla sin contraseña (poco seguro) o establecer la contraseña que desees.

Te pedirá que confirmes la contraseña y a continuación ya se crearán las claves (una pública y una privada).

```
Nuria López Álvarez@NuPC MINGW64 /c
$ ssh-keygen -t ed25519 -C tudireccion@de.correo
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Nuria López Álvarez/.ssh/id_ed25519):
Enter passphrase for "/c/Users/Nuria López Álvarez/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Nuria López Álvarez/.ssh/id_ed25519
Your public key has been saved in /c/Users/Nuria López Álvarez/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:iKY8TS5G6imdJ8w1a6Wp1K1yP8ABbd10Hs9eSd03jfQ tudireccion@de.correo
The key's randomart image is:
+--[ED25519 256]--+
| . o . o . . . . . |
| .o oo + . . . .+ |
| .. .. o o . E |
| . . . . . |
| . . . . .S |
| +oo . |
| .+o+* |
| +oo+B |
| .o*.. |
|-----[SHA256]-----+
```

Sitúate (con el explorador de archivos) en la carpeta donde se han creado las claves:



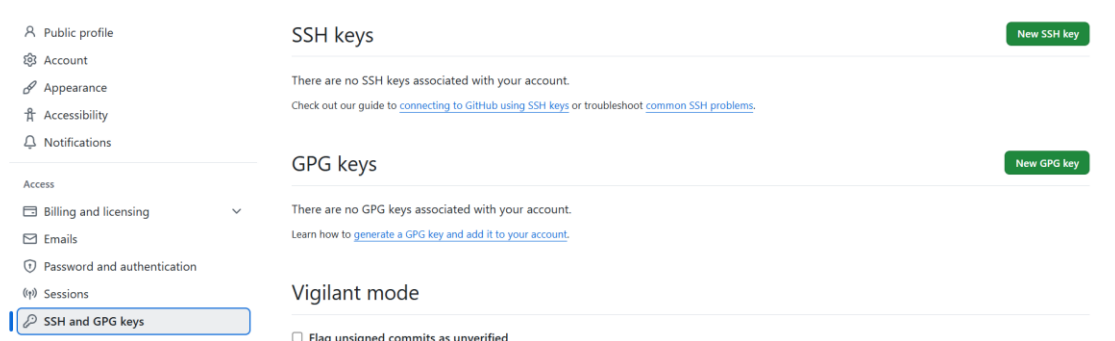
La clave pública es la de extensión .pub, seleccionada en la imagen anterior.

Abrimos el archivo con el bloc de notas y seleccionamos y copiamos todo el contenido:

```
id_ed25519.pub
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIBoRJz3l3uawW/xpHuSvR1Yu5KNq1Qh
bqs5xsYPKhMAC tudireccion@de.correo
```

Vamos a publicarlo en nuestra cuenta de GitHub.

Vamos a Settings de nuestra cuenta de GitHub y seleccionamos “SSH and GPG keys” en el panel izquierdo:



Pulsamos el botón New SSH key

Add new SSH Key

Title

NombreClave

Key type

Authentication Key

Key

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBoRJz3l3uawW/xpHuSvR1Yu5KNq1Qhbqs5xsYPKhMAC tudireccion@de.correo
```

Add SSH key

El título es un identificador para la clave (ponemos lo que queramos). En el recuadro Key pegamos la clave que hemos copiado desde el bloc de notas (.pub) y pulsamos el botón “Add SSH key”.

Ya podemos salir de GitHub y usar el protocolo ssh para manejar nuestros repositorios.

A partir de ahora, cuando git te pida una clave, deberás introducir la que le has asignado a la clave ssh privada durante su creación (en el caso de que no la hubieses dejado en blanco).

NOTA:

Nada más instalar git, deberíamos haber configurado el usuario y su email:

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```