

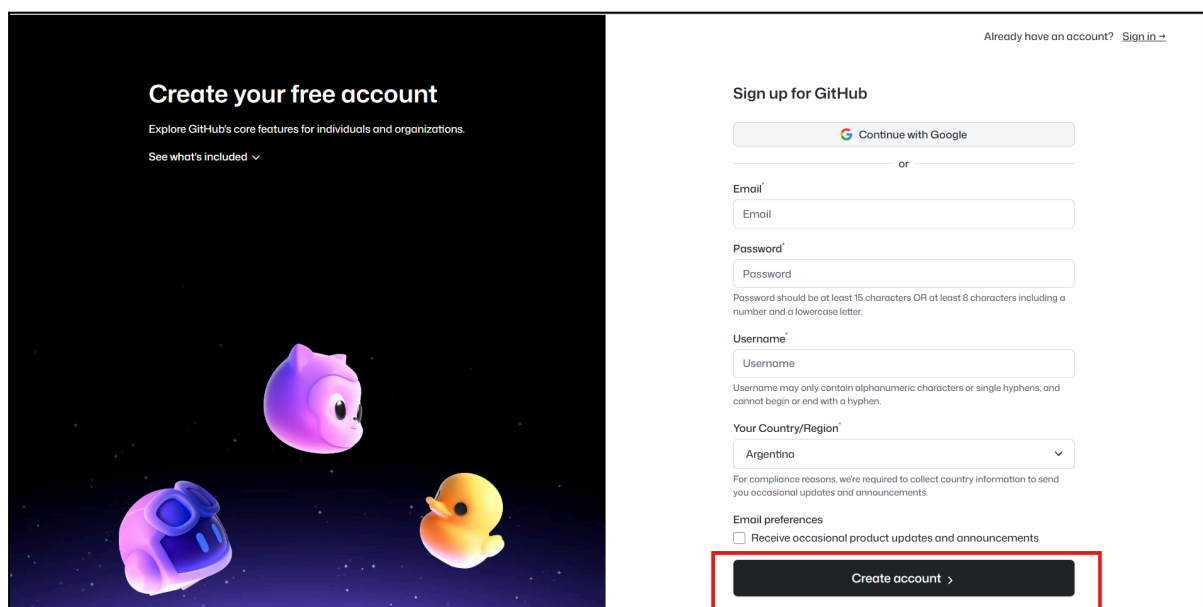
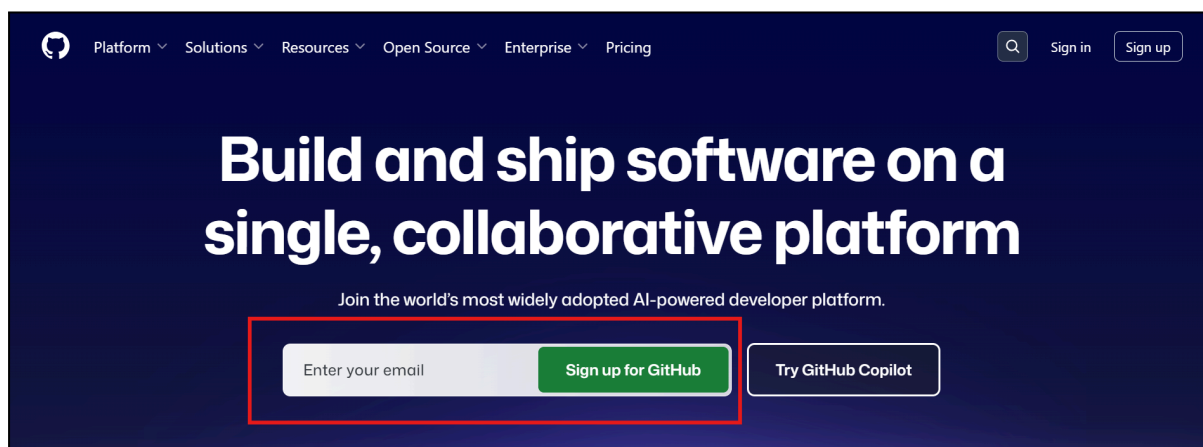
INSTRUCTIVO GIT Y GITHUB

En el presente documento, se compilaron los obstáculos más frecuentes y sus posibles soluciones al momento de utilizar Git y GitHub para el trabajo colaborativo en proyectos de RStudio.

1- CREAR USUARIO EN GITHUB

a- Ingresar a [GitHub.com](https://github.com)

Hacer click en el botón “Sign up for GitHub”, completar los campos solicitados y hacer click en “Create account”.



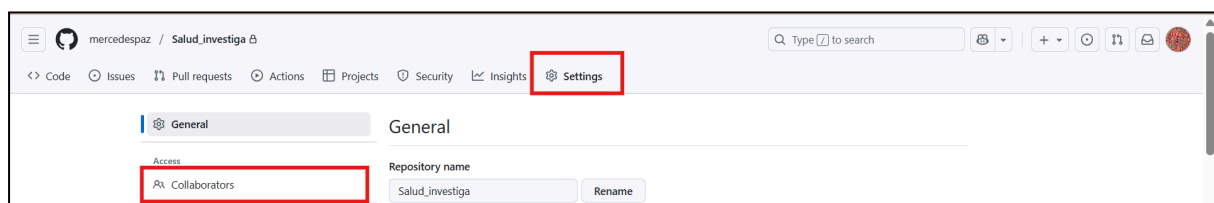
2- CREAR UN REPOSITORIO REMOTO EN GITHUB

Al crear un repositorio remoto podemos configurarlo como público o privado. Si un repositorio es público, cualquier usuario puede acceder al mismo y clonarlo en su computadora. Por el contrario, si es privado, sólo quienes hayan sido invitados a colaborar pueden hacerlo.

Aclaración importante: aunque un repositorio esté configurado como público, el propietario del mismo debe invitar a sus compañeros de equipo a colaborar para asegurarse que puedan hacer modificaciones sin inconvenientes.

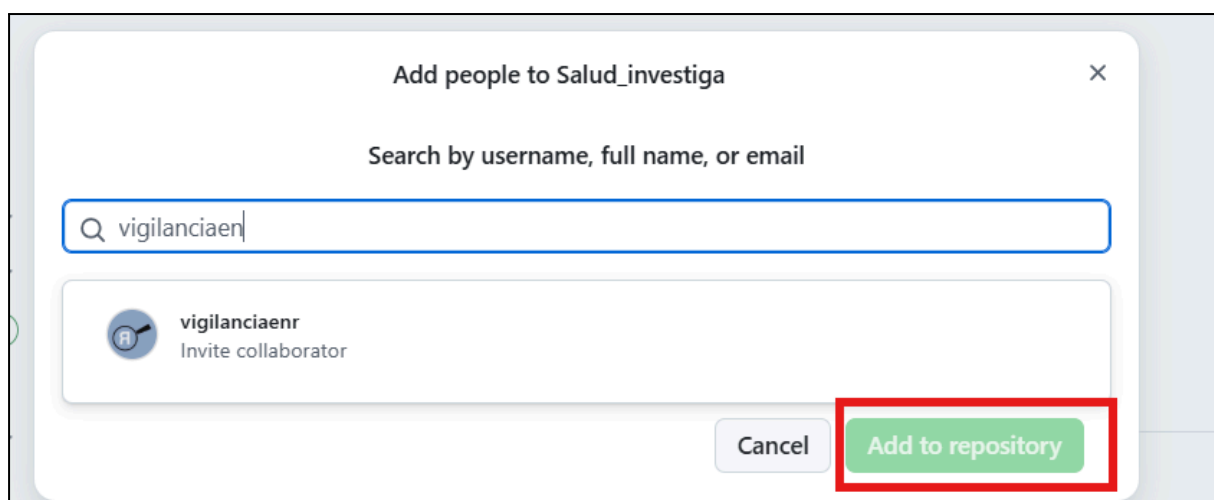
Repositorio privado

Si al crear nuestro repositorio decidimos que fuera privado, tenemos que invitar como colaboradores a los demás integrantes del equipo para que puedan hacer sus contribuciones. Para ello, en el nombre del repositorio hacer click en “Settings” o “Configuración” y buscar en la barra lateral “Collaborators” o “Colaboradores”.



Una vez allí, hacer click en “Add collaborators”. En este paso, escribimos los nombres de usuario de GitHub de las personas que queremos invitar a colaborar y hacemos click en “Add to repository”.

Aclaración: para poder invitar a alguien a colaborar en un repositorio, es necesario que esa persona ya esté registrado con un usuario en GitHub.

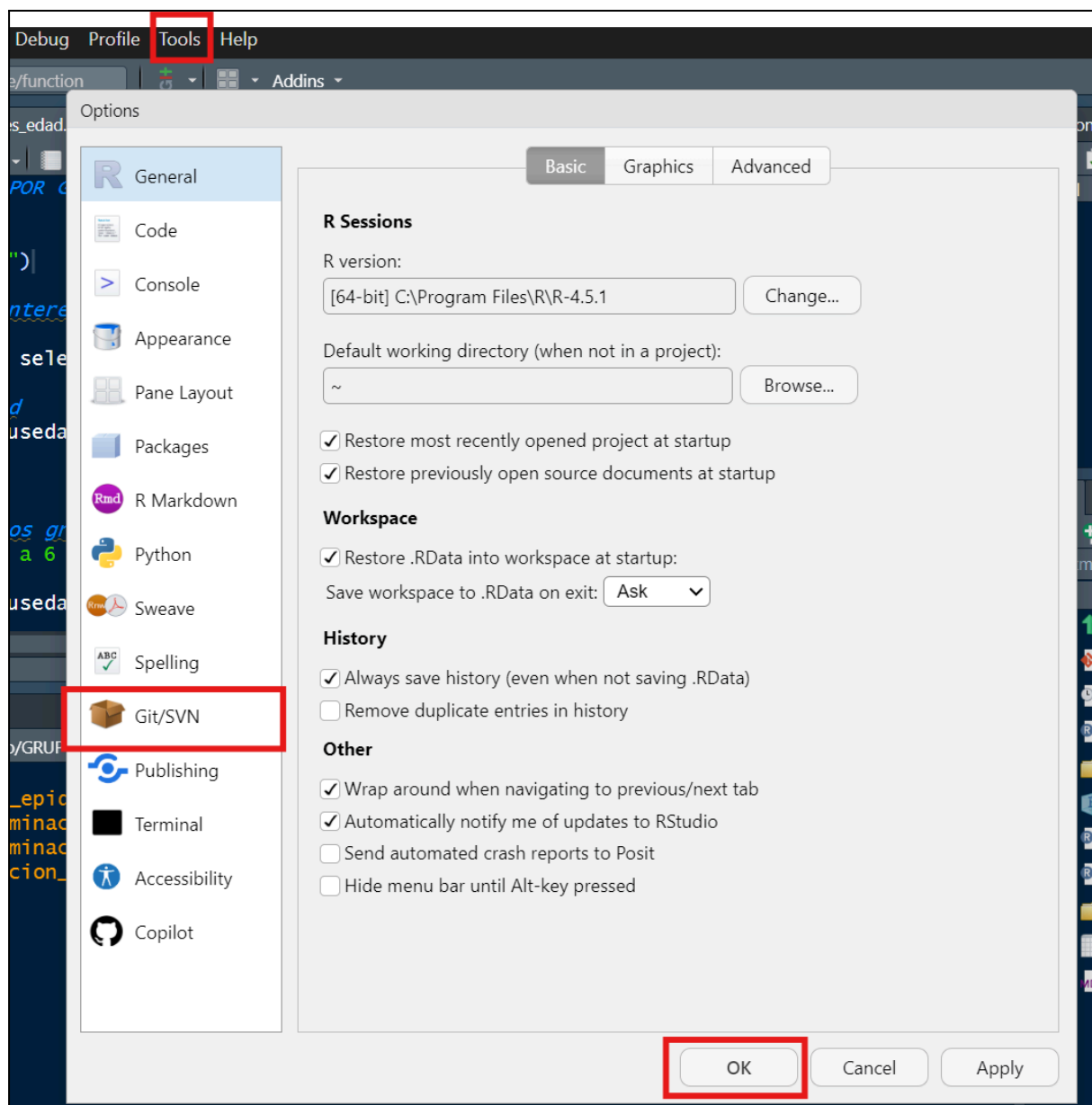


Los usuarios añadidos a colaborar recibirán un mail con la invitación a colaborar, con una validez de 7 días, una vez que la hayan aceptado, podrán sumar su trabajo al repositorio.

3 - CLONAR UN REPOSITORIO REMOTO

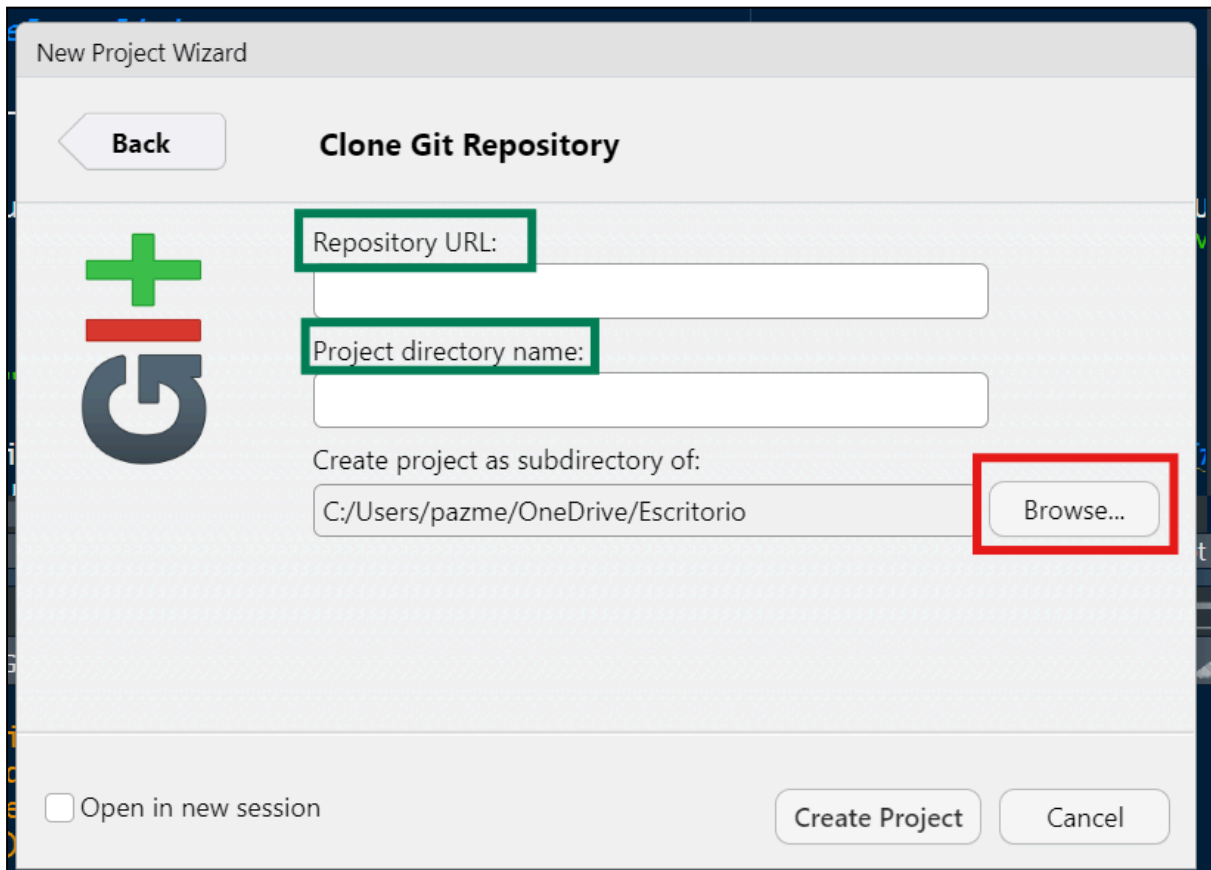
Al clonar un repositorio remoto, es decir, importarlo desde GitHub a nuestra computadora de trabajo, se recomienda prestar atención a los siguientes detalles:

- Verificar haber realizado la conexión de RStudio con Git, tal como se muestra en la siguiente imagen:



- Verificar que se esté asignando una carpeta de directorio al proyecto y que el nombre de la carpeta no exista previamente.

Todos los campos de la imagen siguiente deben estar completos. Para modificar o asignar una carpeta específica en la que se guardará el proyecto, puedo usar el botón "Browse".



4- CÓMO "PRESENTARSE" EN GIT-HUB - AUTHOR IDENTITY UNKNOWN

Esta advertencia puede aparecer al hacer un primer commit en un 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. CSDN @北城笑笑
```

Para solucionarlo podemos realizar lo siguiente:

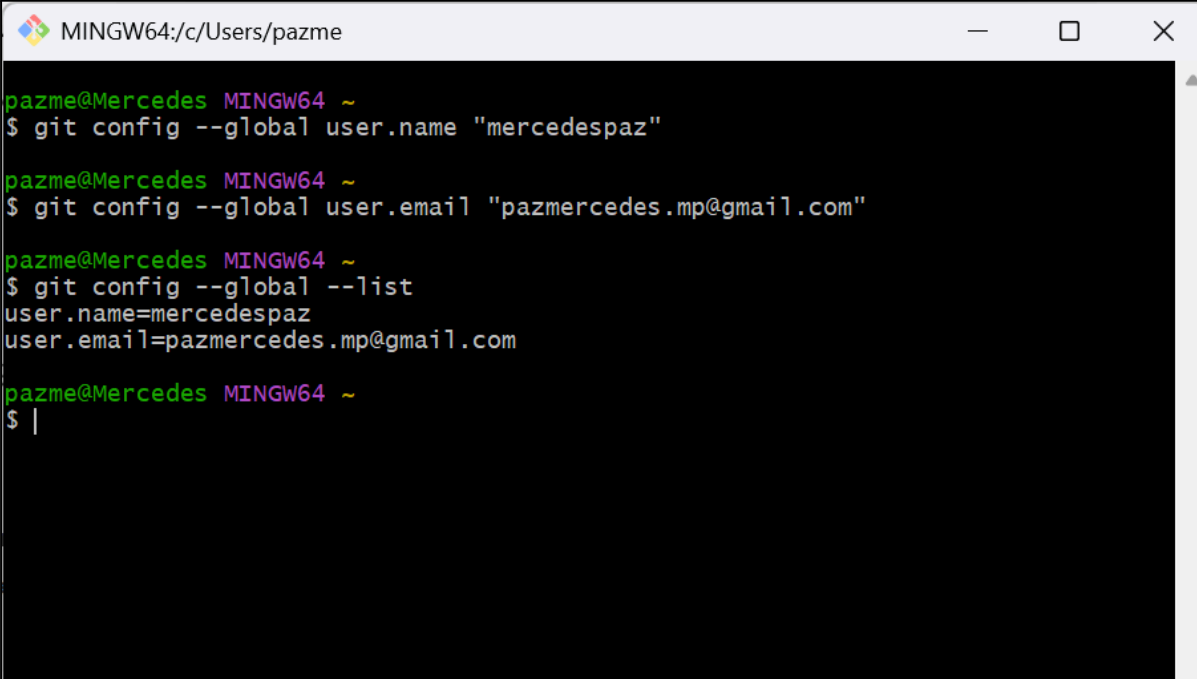
En el buscador de programas de nuestra computadora, buscar Git Bash. Se abrirá una consola en la que hay que ejecutar los siguientes comandos:

git config --global user.name "Tu usuario de GitHub"

git config --global user.email "tumail@ejemplo.com"

Es importante respetar los espacios y signos de puntuación mostrados. El mail a completar, tiene que ser el mail con el que nos registramos al crear la cuenta en GitHub.

La consola de GitBash con los datos ingresados se ve así:

A screenshot of a Git Bash terminal window. The title bar shows the path 'MINGW64:/c/Users/pazme'. The terminal content shows the following commands and output:

```
pazme@Mercedes MINGW64 ~  
$ git config --global user.name "mercedespaz"  
  
pazme@Mercedes MINGW64 ~  
$ git config --global user.email "pazmercedes.mp@gmail.com"  
  
pazme@Mercedes MINGW64 ~  
$ git config --global --list  
user.name=mercedespaz  
user.email=pazmercedes.mp@gmail.com  
  
pazme@Mercedes MINGW64 ~  
$ |
```

El comando git config --global --list confirma el registro.

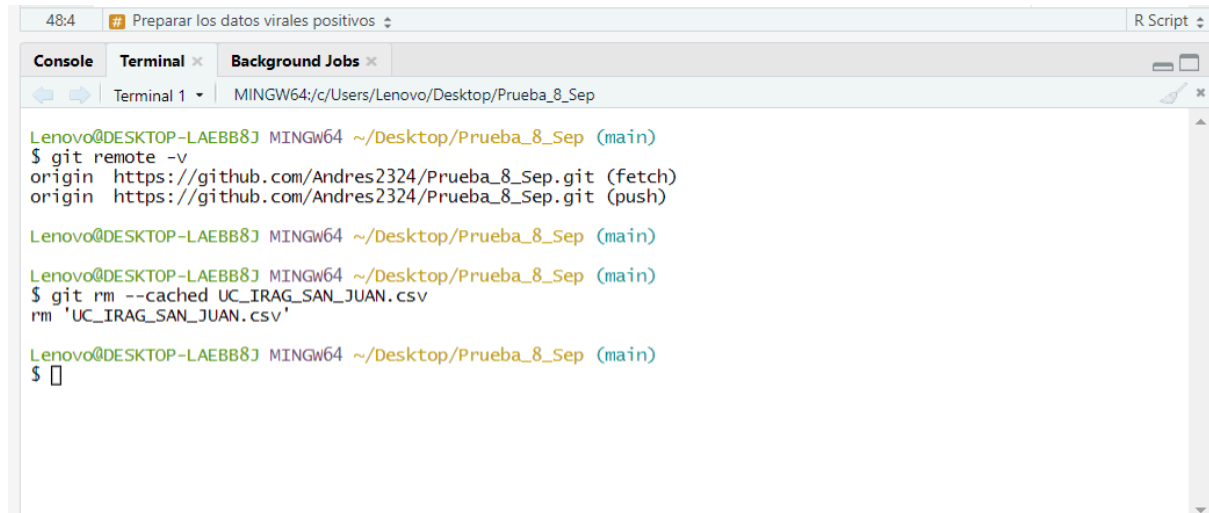
5- ELIMINAR UNA BASE DE DATOS DE UN REPOSITORIO PÚBLICO EN GITHUB.

Puede pasar que cuando creamos nuestro repositorio, luego lo clonamos en nuestra PC y nos ponemos a trabajar en nuestro proyecto.

Sin embargo a la hora de subirlo nuevamente a GitHub notamos que subimos a nuestro repositorio online nuestra base de datos, que contiene información que no queremos que sea pública.

Para resolver esto tenemos que eliminar esa base de datos de nuestro repositorio online en GitHub, pero queremos que permanezca en nuestro repositorio local en nuestra computadora.

Para eliminar la base de datos en nuestro repositorio en GitHub, tenemos que escribir en RStudio en la terminal



```
48:4 Preparar los datos virales positivos R Script
Terminal 1 MINGW64/c/Users/Lenovo/Desktop/Prueba_8_Sep
Lenovo@DESKTOP-LAEBB8J MINGW64 ~/Desktop/Prueba_8_Sep (main)
$ git remote -v
origin https://github.com/Andres2324/Prueba_8_Sep.git (fetch)
origin https://github.com/Andres2324/Prueba_8_Sep.git (push)
Lenovo@DESKTOP-LAEBB8J MINGW64 ~/Desktop/Prueba_8_Sep (main)
$ git rm --cached UC_IRAG_SAN_JUAN.csv
rm 'UC_IRAG_SAN_JUAN.csv'
Lenovo@DESKTOP-LAEBB8J MINGW64 ~/Desktop/Prueba_8_Sep (main)
$
```

a) Borrarlo del repositorio. Escribir:

`git rm --cached nombredelabase.csv`

El `--cached` le dice a Git que lo elimine del repo pero no de tu compu (queda en tu carpeta local).

b) Commit del cambio

`git commit -m "Eliminar base de datos del repositorio"`

c) Push

`git push origin main`

En caso de que no quiera que se suban este tipo de archivos a un repositorio público pero si actualizar los scripts del proyecto, tengo que colocar estos en el archivo `.gitignore`

	Name	Size	Modified
	..		
<input type="checkbox"/>	 .gitignore	30 B	Sep 8, 2025, 6:19 PM

Los archivos `.gitignore` pueden crearse. Son archivos de texto "Text File" y pueden crearse desde la ventana Files. En esos archivos de textos escribimos lo que queremos ignorar a la hora de hacer commit y luego lo guardamos con el nombre `.gitignore`.

Si tenemos más de una base o archivo podemos agregarlos en el .gitignore de la siguiente forma:

```
# Archivos de datos sensibles
```

```
datos.csv
```

```
*.xlsx
```

```
# Archivos temporales de R
```

```
.Rhistory
```

```
.RData
```

```
.Rproj.user/
```

```
# Archivos de Quarto renderizados
```

```
*.html
```

```
*.pdf
```

Los archivos listados en [.gitignore](#) ya no deberían aparecer como cambios a commitear. Con el símbolo * damos la orden de que se ignoren todos los archivos, en caso de que tengamos mas de una base de datos con información sensible.

6- ERROR 403 “UNABLE TO ACCESS”

Este error puede presentarse al hacer “Push”, es decir, enviar nuestros cambios del repositorio local al remoto.



En esta situación podemos:

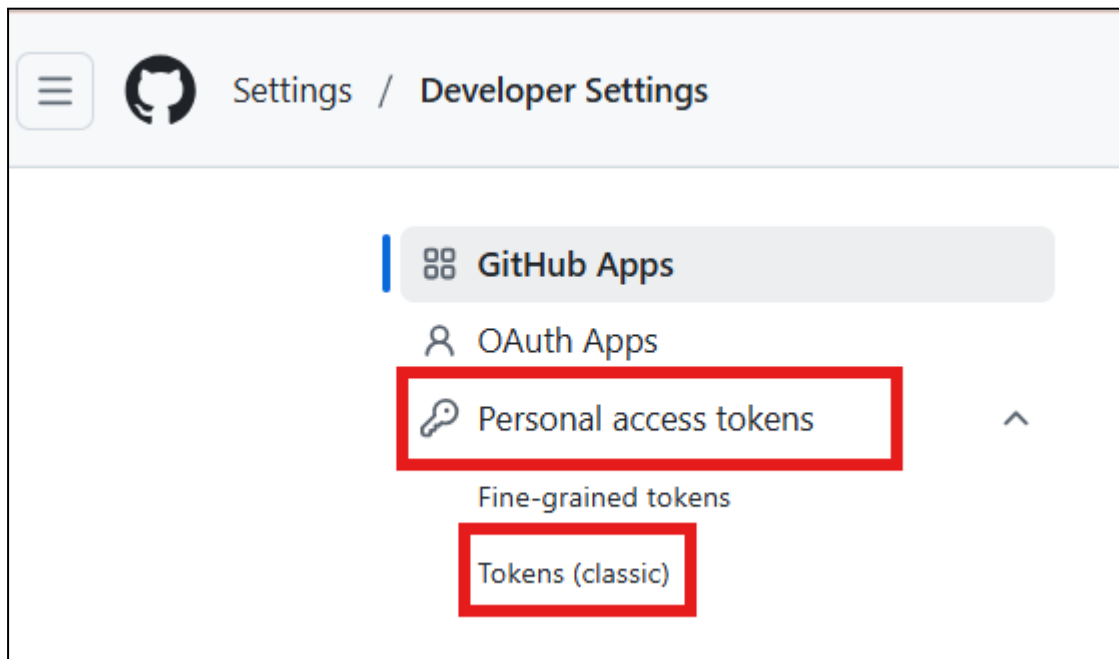
- En primera instancia, asegurarnos de haber sido invitados a colaborar en el proyecto para poder editar el mismo.

- Verificar que nuestra clave de autenticación “Token” sea válida y vigente. En algunas situaciones, GitHub puede pedirnos una clave al momento de hacer “Push” en un proyecto.

7- CREAR CLAVE TOKEN

Ingresa a nuestra cuenta de GitHub, y hacer click en el ícono de nuestro usuario. Allí seleccionar “Settings” o “Configuración”.

En la ventana de trabajo que aparece, buscar en el margen inferior izquierdo la opción “Developer settings”. Luego, clickear las opciones marcadas a continuación:



Luego hacer click en “Generate new token” y seleccionar “Generate new token (classic)”. Designar un nombre al token y configurar una fecha de expiración para el mismo.

Seleccionar los alcances o “scopes” de la clave, mínimamente seleccionar “repo” para acceder y colaborar en repositorios.

Hacer click en “Generate token” copiar el código generado.

Importante: El código token se muestra una única vez, por lo que se sugiere copiarlo y guardarlo en un archivo aparte.

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_1234567890abcdefghijklmnopqrstuvwxyz2 [Delete](#)

Personal access tokens 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](#).

Luego de generada la clave token, al intentar hacer “Push”, se abrirá una ventana como la siguiente en la que se selecciona la opción “Token” y se pega la clave generada por única vez.

