

## IMPLEMENTACIÓN DE GIT Y GITHUB

Instituto Superior Tecnológico CENESTUR

Bryan D. Velastegui L.

Lenin J. Guachamin R.

Shynlu P. Chimbo T.

Desarrollo de Software - 2025.2

Programación web para ciencia de datos / Segundo Bloque

Msc. Cristian David Muñoz Tenempaguay

12 de diciembre de 2025

<b>IMPLEMENTACIÓN DE GIT Y GITHUB.....</b>	<b>1</b>
<b>Objetivo.....</b>	<b>3</b>
<b>Configuración Git.....</b>	<b>3</b>
Instalación.....	3
Configuración.....	5
Práctica.....	6
<b>Configuración GitHub.....</b>	<b>10</b>
Creación de cuenta.....	10
Creación y configuración de clave ssh.....	11
Creación de un repositorio.....	14
Subir archivos al repositorio.....	15
<b>Enlaces de repositorios.....</b>	<b>16</b>
<b>Conclusiones.....</b>	<b>17</b>
<b>Referencias.....</b>	<b>17</b>

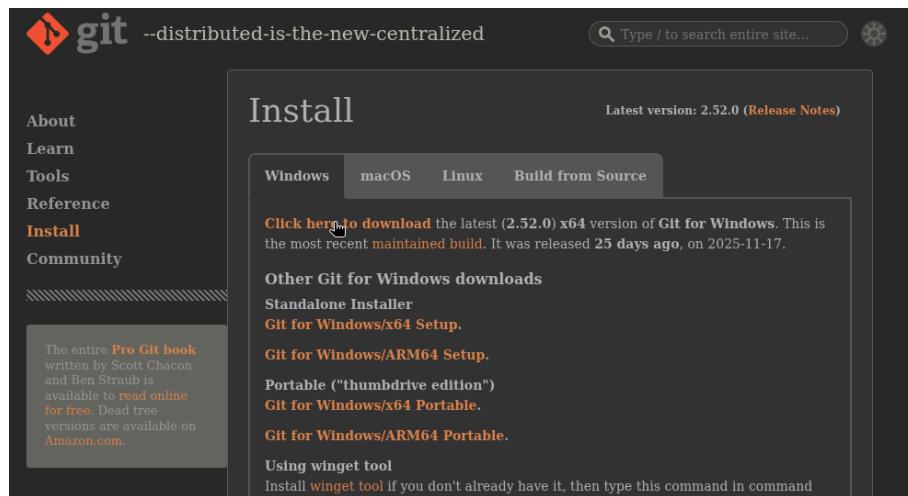
## Objetivo

Configurar Git y Github

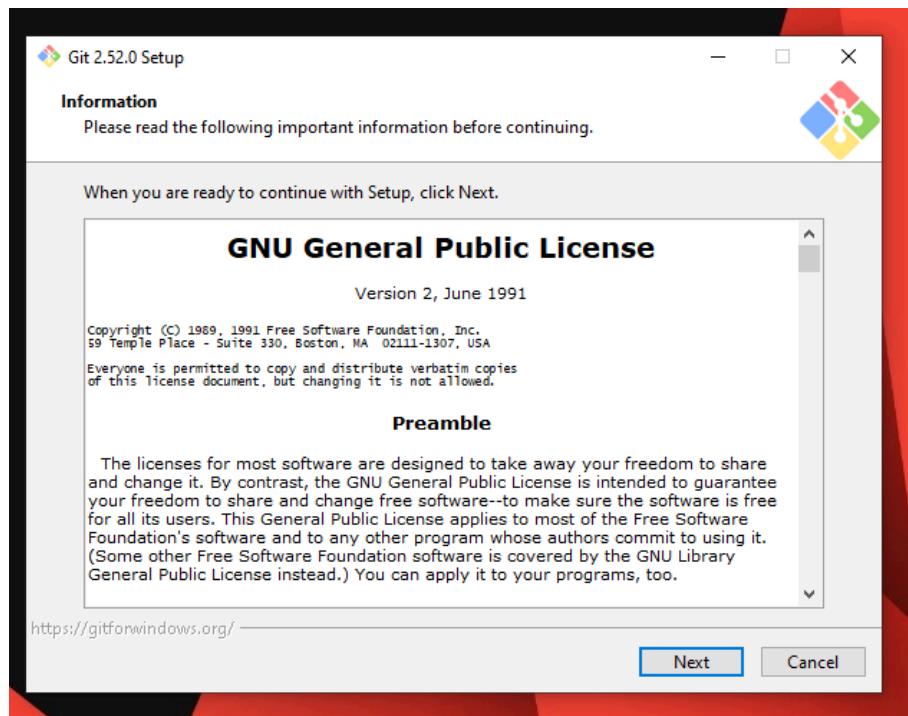
### Configuración Git

#### Instalación

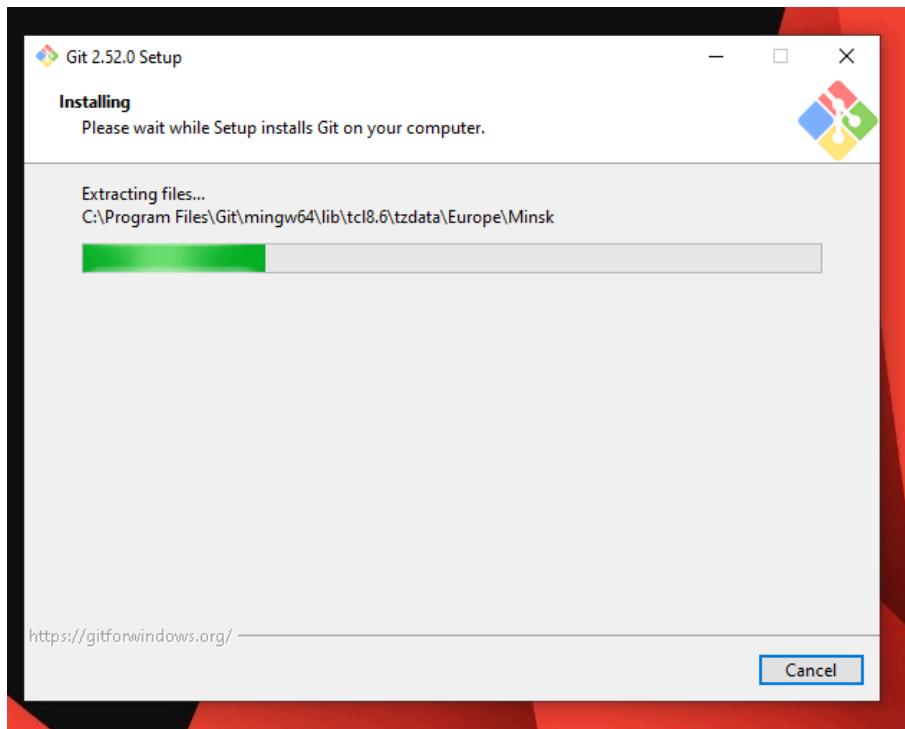
Descargamos el instalador desde la página web de git: <https://git-scm.com/install/windows>



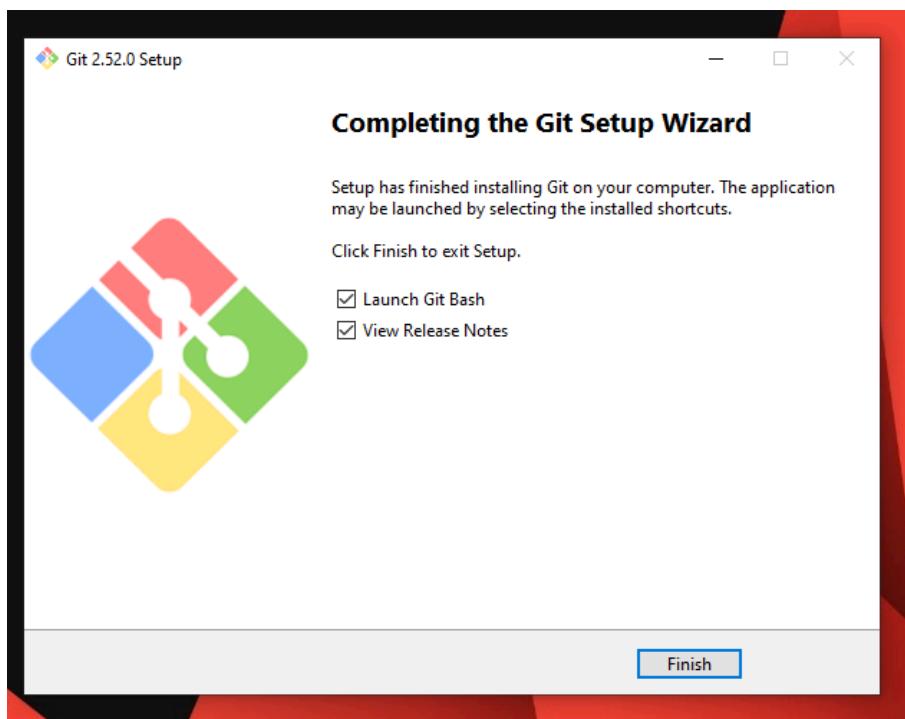
Al completar la descarga ejecutamos el instalador



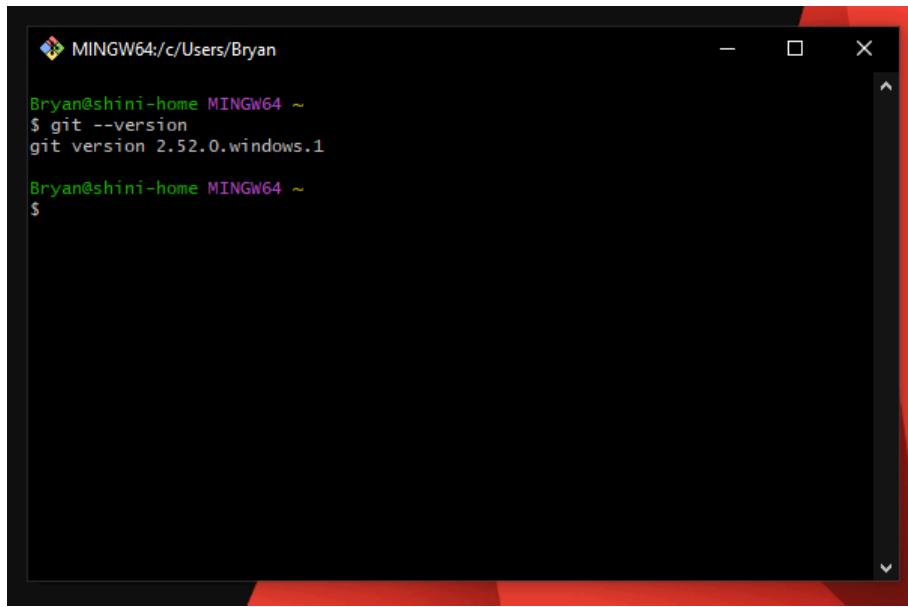
Utilizamos todas configuraciones por defecto y procedemos con la instalación



Al finalizar la instalación marcamos la casilla *Launch Git Bash* para verificar la instalación, y presionamos **Finish**.



Ejecutamos el comando `git --version` y podemos confirmar que la instalación se realizó con éxito.



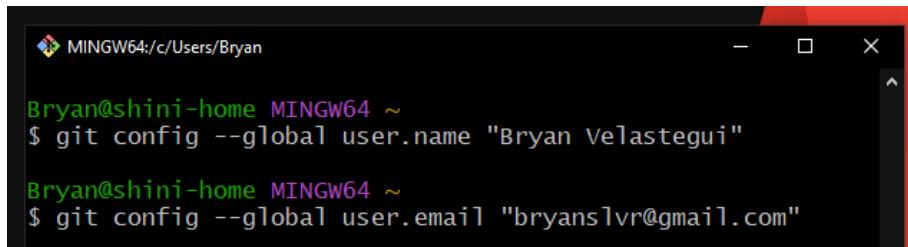
```
MINGW64:/c/Users/Bryan
Bryan@shini-home MINGW64 ~
$ git --version
git version 2.52.0.windows.1
Bryan@shini-home MINGW64 ~
$
```

## Configuración

Configuramos el nombre y el correo que se usarán para las confirmaciones (*commit*) de git con los siguientes comandos:

```
git config --global user.name 'Bryan Velastegui'
```

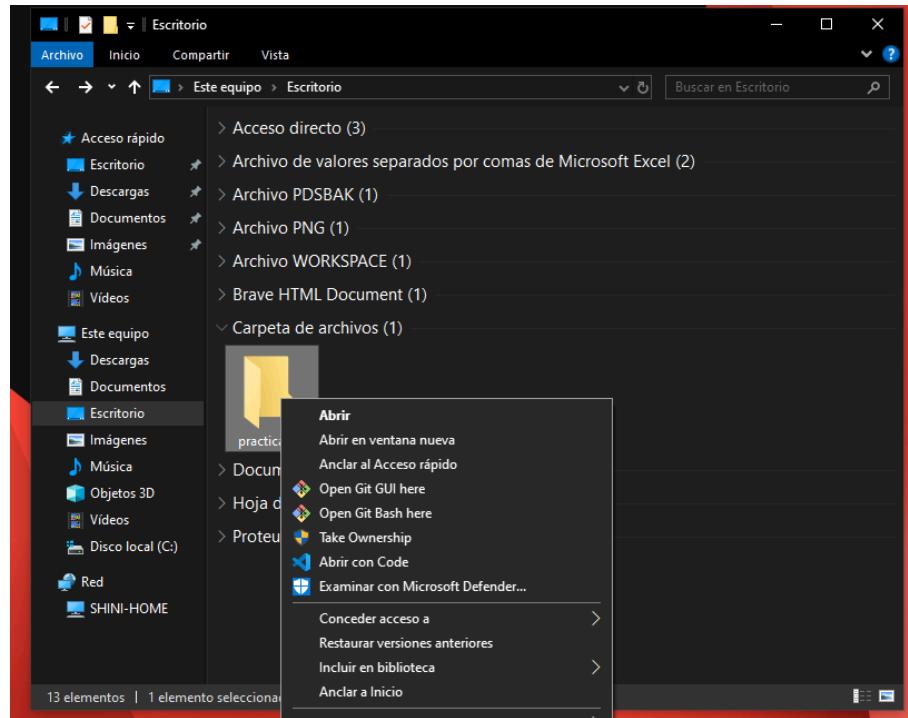
```
git config --global user.email 'bryanslvr@gmail.com'
```



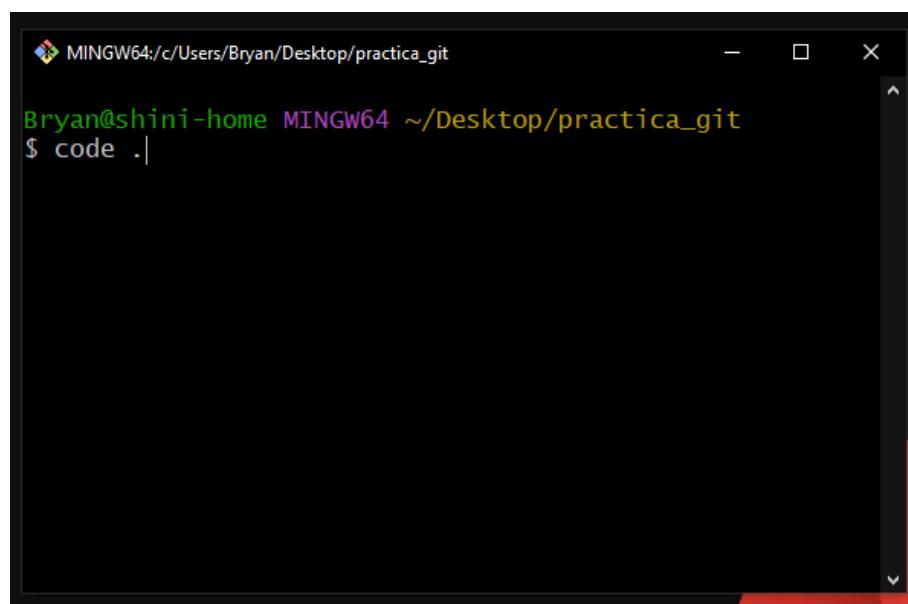
```
MINGW64:/c/Users/Bryan
Bryan@shini-home MINGW64 ~
$ git config --global user.name "Bryan Velastegui"
Bryan@shini-home MINGW64 ~
$ git config --global user.email "bryanslvr@gmail.com"
```

## Práctica

En el escritorio creamos una carpeta llamada `practica_git`, presionamos click derecho sobre la carpeta y en el menú contextual damos click en `Open Git Bash Here`

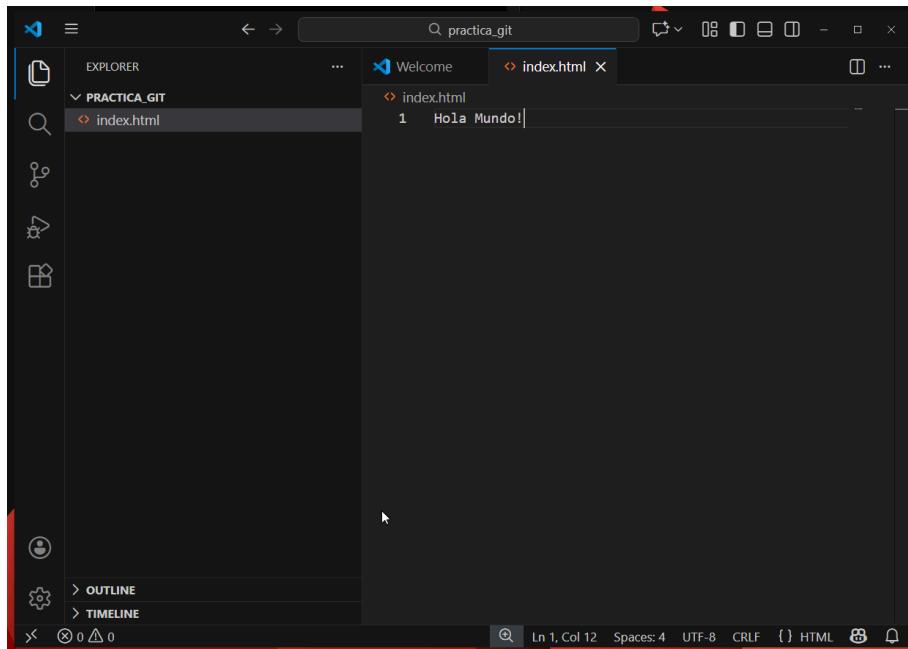


En la consola usamos el comando `code .` para abrir nuestra carpeta usando Visual Studio Code:

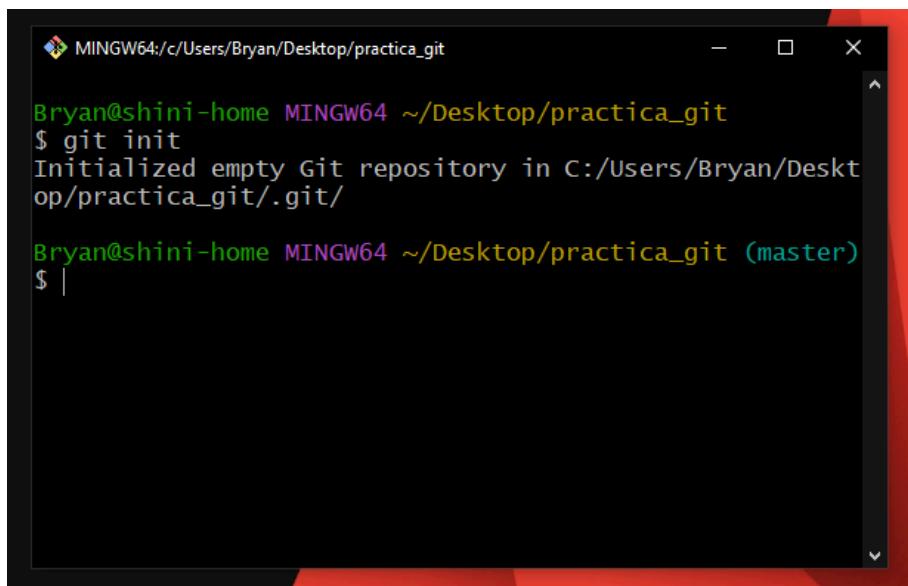


```
MINGW64:c/Users/Bryan/Desktop/practica_git
Bryan@shini-home MINGW64 ~/Desktop/practica_git
$ code .|
```

En vs code creamos un nuevo archivo llamado index.html y agregamos hola mundo en su contenido.

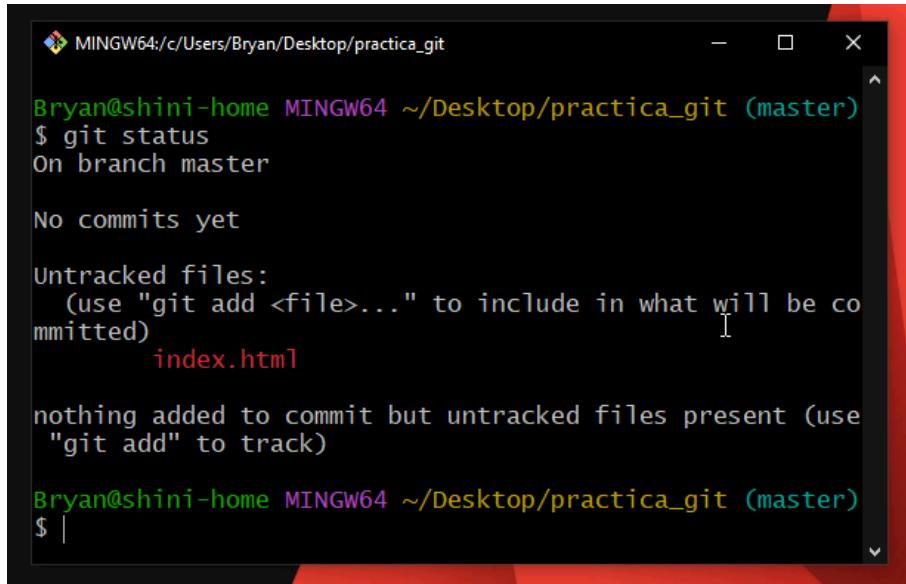


En la consola usamos el comando `git init` para inicializar un repositorio vacío de git:



```
Bryan@shini-home MINGW64 ~/Desktop/practica_git
$ git init
Initialized empty Git repository in C:/Users/Bryan/Desktop/practica_git/.git/
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ |
```

Con el comando `git status` podemos notar que nuestro archivo `index.html` está creado pero no se le da seguimiento:



```
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git status
On branch master

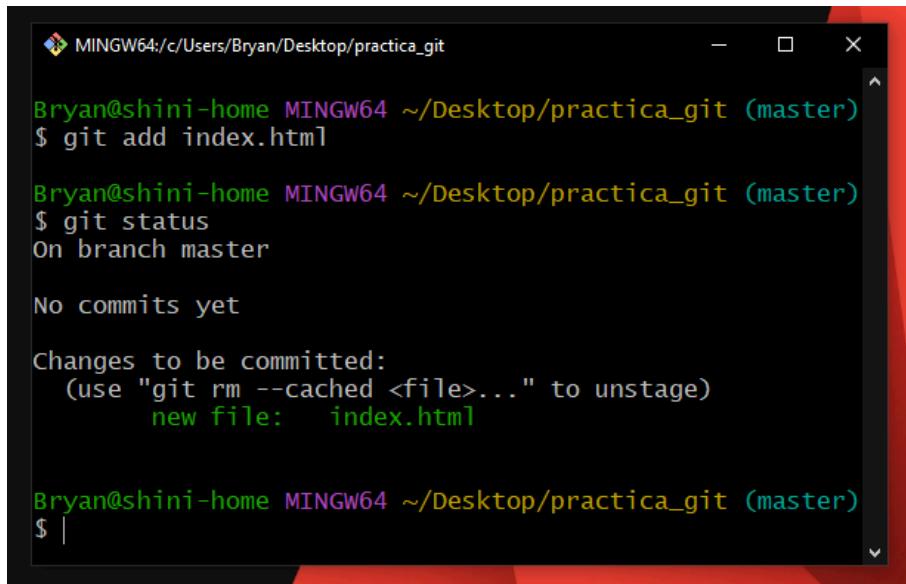
No commits yet

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)

Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ |
```

Para darle seguimiento usamos el comando `git add index.html` seguido de un `git status` y se puede notar que los cambios de nuestro archivo se registraron en el escenario actual (stage)



```
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git add index.html

Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git status
On branch master

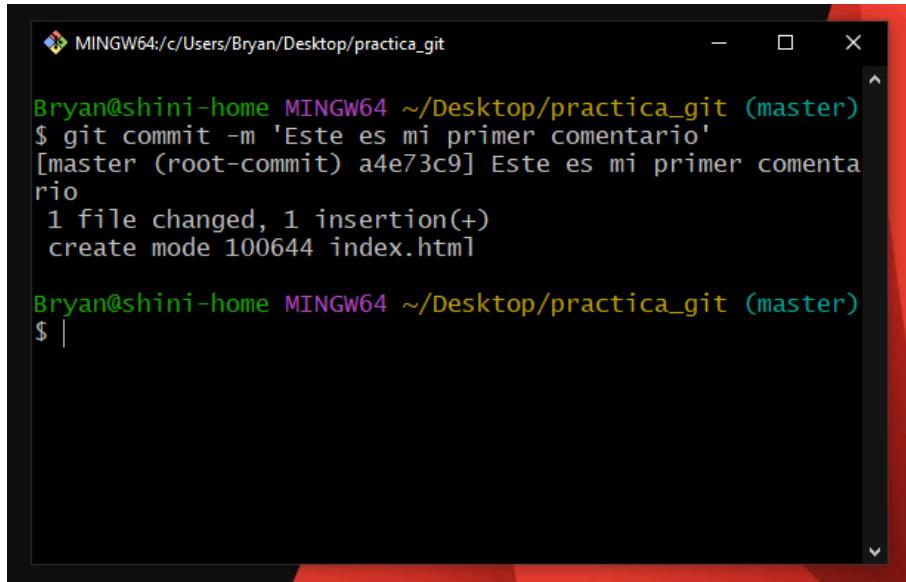
No commits yet

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

Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ |
```

Teniendo en cuenta que git es un sistema de control de versiones, para cada cambio debemos confirmar el asentamiento de los mismos.

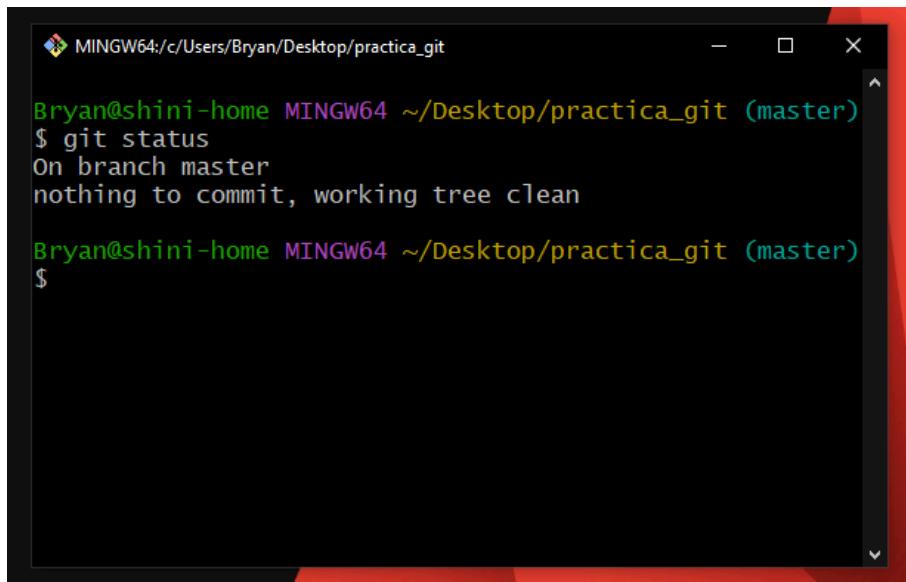
Utilizamos el comando `git commit -m 'Este es mi primer cambio'` para confirmar y asentar en el repositorio los cambios del archivo index.html



```
MINGW64:/c/Users/Bryan/Desktop/practica_git
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git commit -m 'Este es mi primer comentario'
[master (root-commit) a4e73c9] Este es mi primer comentario
 1 file changed, 1 insertion(+)
 create mode 100644 index.html

Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ |
```

Si volvemos a ejecutar el comando `git status` nos indica que nuestro árbol de trabajo está limpio.

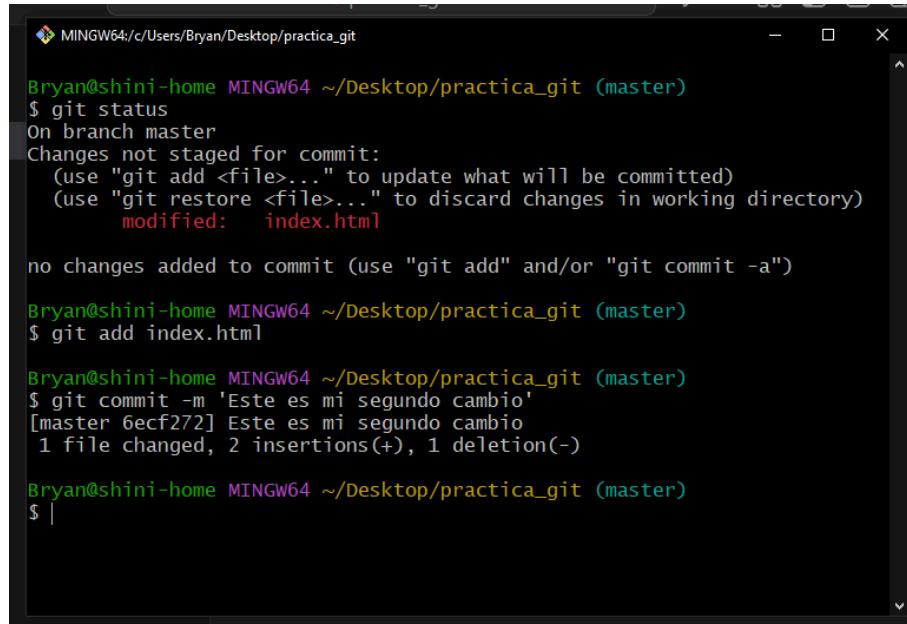


```
MINGW64:/c/Users/Bryan/Desktop/practica_git
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git status
On branch master
nothing to commit, working tree clean

Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$
```

Si modificamos el archivo index.html el comando `git status` nos indica que el archivo fue modificado y para confirmar los cambios realizados nuevamente utilizamos los comandos:

1. `git add index.html` Agregamos los cambios al escenario
2. `git commit -m 'Este es mi segundo cambio'` Confirmamos los cambios y asentamos en el repositorio de git



```
MINGW64:/c/Users/Bryan/Desktop/practica_git
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git add index.html

Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git commit -m 'Este es mi segundo cambio'
[master 6ecf272] Este es mi segundo cambio
 1 file changed, 2 insertions(+), 1 deletion(-)

Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ |
```

## Configuración GitHub

GitHub es una plataforma basada en la nube donde podemos almacenar y compartir nuestros repositorios de Git.

## Creación de cuenta

Para crear una cuenta debemos dirigirnos a la página <https://github.com/signup> y completar la información requerida, siempre se recomienda usar la cuenta de correo personal, nunca se debe usar una cuenta de correo institucional o laboral a menos que sea necesario.

Esta cuenta de correo debe ser igual a la que se utilizó en la [Configuración de Git](#).

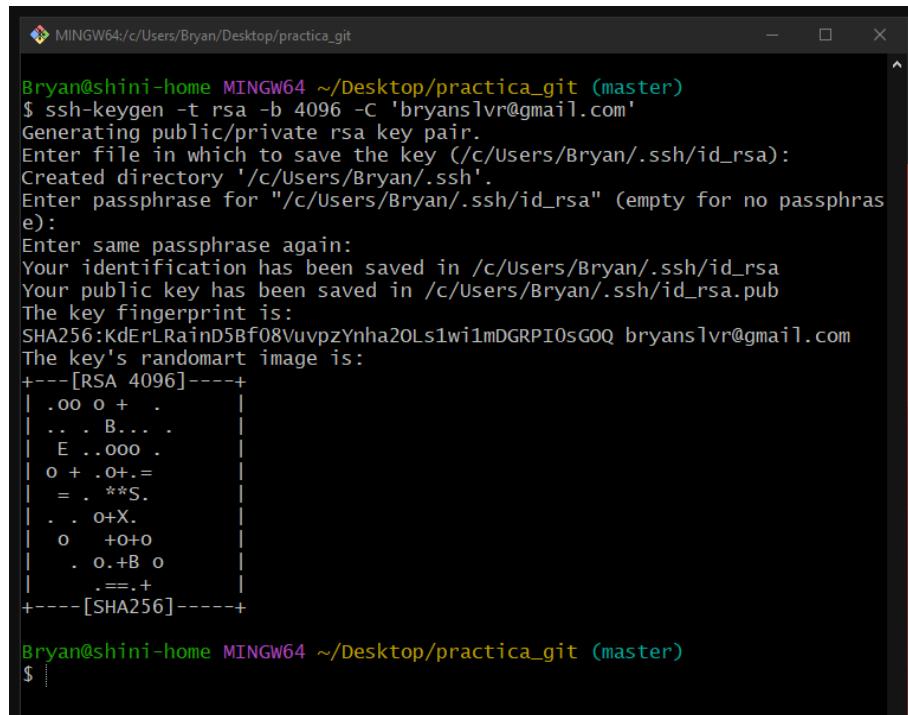
## Creación y configuración de clave ssh

Para poder subir nuestros repositorios de Git en Github es necesario agregar una clave ssh, a continuación vamos a indicar los pasos necesarios para crear una clave ssh y configurarla en nuestra cuenta de GitHub.

En nuestro Git Bash ejecutamos el comando:

```
ssh-keygen -t rsa -b 4096 -C 'bryanslvr@gmail.com'
```

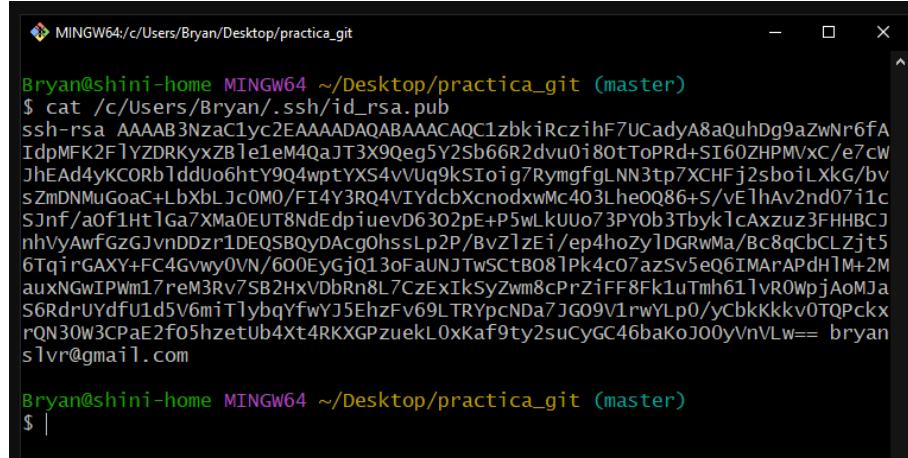
Este comando genera una clave ssh que usa el algoritmo RSA (-t rsa) con una cantidad de 4096 bits (-b 4096) y con un comentario que hace referencia a nuestro correo electrónico (-C '[bryanslvr@gmail.com](mailto:bryanslvr@gmail.com)')



```
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ ssh-keygen -t rsa -b 4096 -C 'bryanslvr@gmail.com'
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Bryan/.ssh/id_rsa):
Created directory '/c/Users/Bryan/.ssh'.
Enter passphrase for "/c/Users/Bryan/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Bryan/.ssh/id_rsa
Your public key has been saved in /c/Users/Bryan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:KdErLRainD5Bf08UvpzYnha20Ls1wi1mDGRPIOsGOQ bryanslvr@gmail.com
The key's randomart image is:
+---[RSA 4096]---+
| .oo o + . |
| ... B.... . |
| E ..ooo . |
| o + .o+= |
| = . **S. |
| . . o+X. |
| o +o+o |
| . o.+B o |
| .==.+ |
+---[SHA256]---+
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$
```

En el resultado del comando podemos observar la ruta de nuestra clave ssh pública `Your`

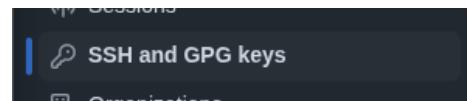
`public key has been saved in /c/Users/Bryan/.ssh/id_rsa.pub` utilizamos el comando cat para ver y poder copiar el contenido de nuestra clave ssh publica:



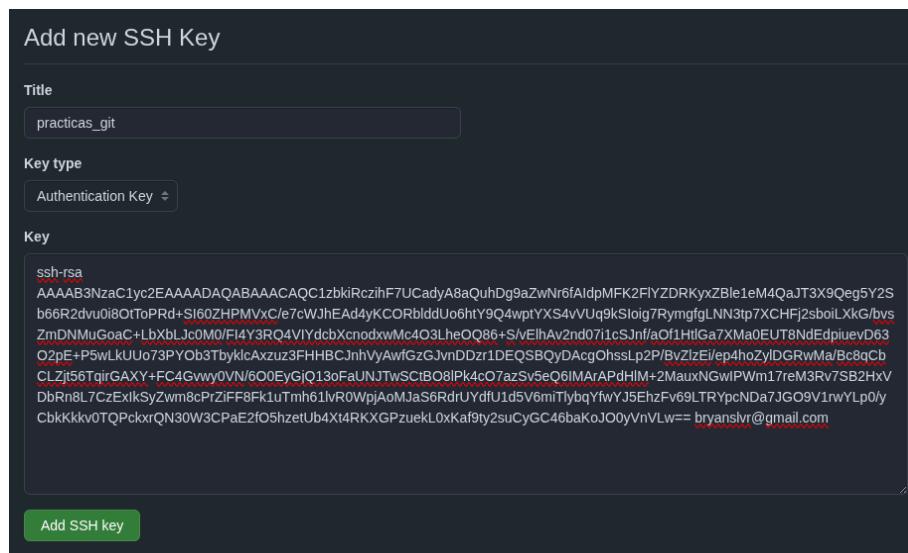
```
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ cat /c/Users/Bryan/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQAC1zbkiRczihF7UCadyA8aQuhDg9aZwNr6fA
IdpMFk2F1YzDRKyxZB1eM4QaJT3X9Qeg5Y2Sb66R2dvu0i80tToPrd+SI60ZHPMVxC/e7cW
JhEAd4yKCOrb1dduo6htY9Q4wpTyxs4vVUq9kS1oig7RymfgLNN3tp7XCHFj2sboiLxkG/bv
sZmDNMuGoaC+lBxbLJcOM0/FI4Y3RQ4VIYdcBxcnodxwMc403LheQ86+S/vElhAv2nd07i1c
Sjnf/aOf1Ht1ga7XMa0EUT8NdEdp1uevD6302pE+P5wLkuUo73PY0b3Tbyk1CAxzuz3FHHBCJ
nhVawfGzGJvnDDzr1DEQSBQyDACg0hssLp2P/BvZlzei/ep4hoZylDGRwMa/Bc8qCbCLzjt5
6TqirGAXY+FC4Gvwy0VN/600EygjQ13oFaUNJTwScTB081Pk4c07azSv5eQ6IMArAPdH1M+2M
auxNGwIPWm17reM3Rv7SB2HxDpRn8L7CzExIkSyZwm8cPrz1FF8Fk1uTmh611vR0WpjAoMJa
S6RdrUYdfu1d5V6miT1ybqYfwJ5EhzFv69LTRYpcNDa7JGO9V1rwYLp0/yCbkKKkv0TQPckx
rQN30W3CPaE2f05hZetUb4Xt4RKXGPzuekL0xKaf9ty2suCyGC46baKoJ00yVnLw== bryan
s1vr@gmail.com
```

Copiamos el contenido de nuestra clave, nos dirigimos a las configuraciones de nuestro

perfil de GitHub, en la sección de SSH and GPG Keys



Presionamos el botón New SSH key, completamos el *Título* y en el cuadro de texto de *Key*, pegamos el contenido de nuestra clave ssh



Add new SSH Key

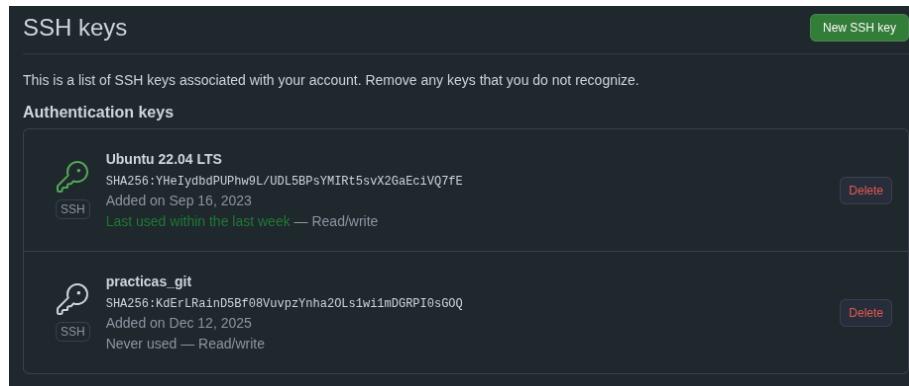
Title  
practicas\_git

Key type  
Authentication Key

Key  
ssh-rsa  
AAAAB3NzaC1yc2EAAAQABAAQAC1zbkiRczihF7UCadyA8aQuhDg9aZwNr6fAIdpMFk2F1YzDRKyxZB1eM4QaJT3X9Qeg5Y2Sb66R2dvu0i80tToPrd+SI60ZHPMVxC/e7cWJhEAd4yKCOrb1dduo6htY9Q4wpTyxs4vVUq9kS1oig7RymfgLNN3tp7XCHFj2sboiLxkG/bvsZmDNMuGoaC+lBxbLJcOM0/FI4Y3RQ4VIYdcBxcnodxwMc403LheQ86+S/vElhAv2nd07i1cSjnf/aOf1Ht1ga7XMa0EUT8NdEdp1uevD63O2pE+P5wLkuUo73PY0b3Tbyk1CAxzuz3FHHBCJnhVawfGzGJvnDDzr1DEQSBQyDACg0hssLp2P/BvZlzei/ep4hoZylDGRwMa/Bc8qCbCLzjt56TqirGAXY+FC4Gvwy0VN/600EygjQ13oFaUNJTwScTB081Pk4c07azSv5eQ6IMArAPdH1M+2MauxNGwIPWm17reM3Rv7SB2HxDpRn8L7CzExIkSyZwm8cPrz1FF8Fk1uTmh611vR0WpjAoMJaS6RdrUYdfu1d5V6miT1ybqYfwJ5EhzFv69LTRYpcNDa7JGO9V1rwYLp0/yCbkKKkv0TQPckxrQN30W3CPaE2f05hZetUb4Xt4RKXGPzuekL0xKaf9ty2suCyGC46baKoJ00yVnLw== bryans1vr@gmail.com

Add SSH key

Al presionar el botón *Add SSH key* regresamos a la lista de llaves y encontraremos la llave que acabamos de registrar (*practicas\_git*)



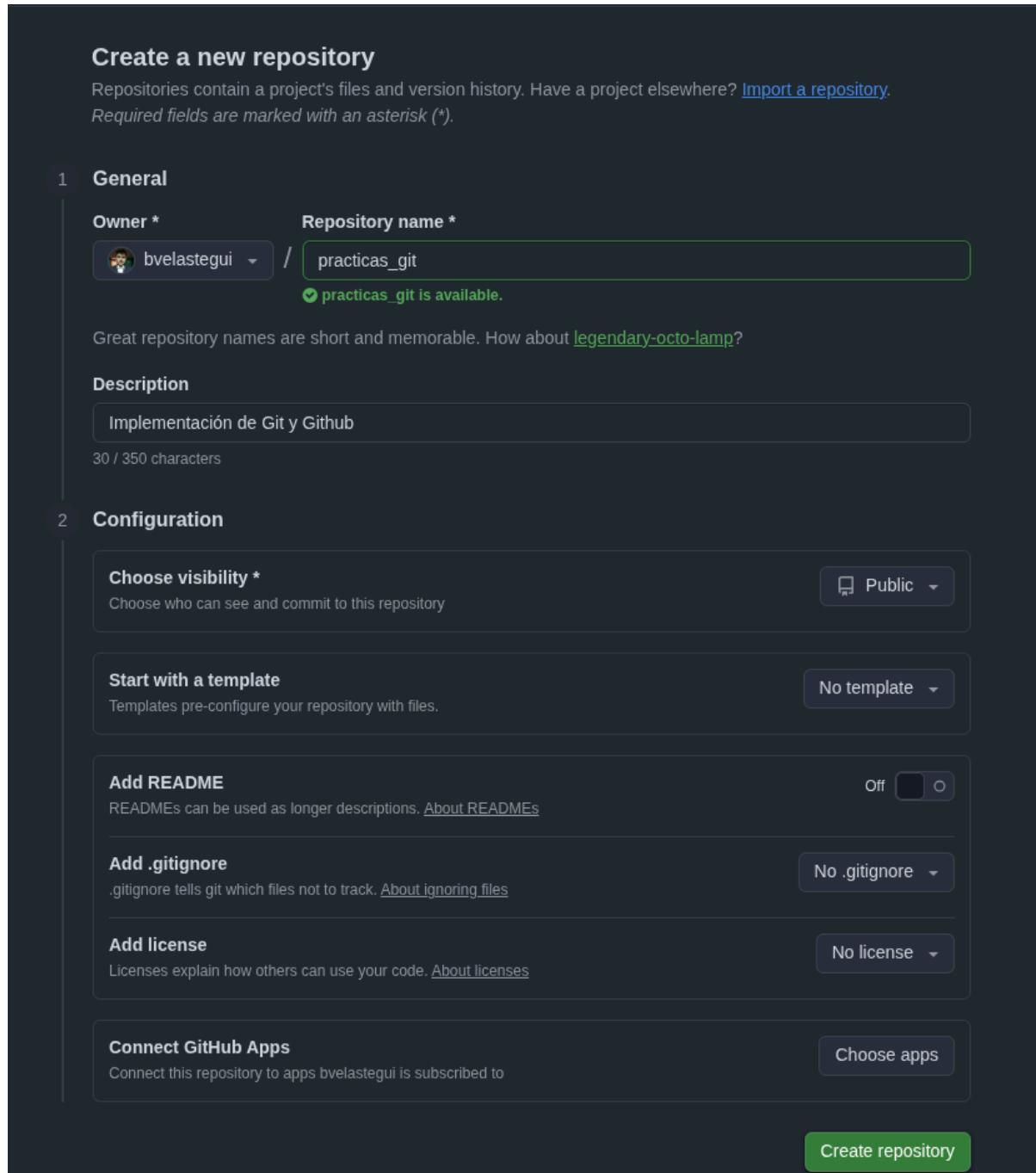
The screenshot shows the 'SSH keys' page from GitHub. At the top, there is a green button labeled 'New SSH key'. Below it, a message says 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Under the heading 'Authentication keys', there are two entries:

- Ubuntu 22.04 LTS**  
SSH icon, SHA256: YHeIydbdPUPhw9L/UDL5BPsYMIRt5svX2GaEc1VQ7fE, Added on Sep 16, 2023, Last used within the last week — Read/write, Delete button.
- practicas\_git**  
SSH icon, SHA256: KdErLRain05Bf08UvvpzYnha20Ls1wi1mDGRPI0sG0Q, Added on Dec 12, 2025, Never used — Read/write, Delete button.

## Creación de un repositorio

Para crear un repositorio podemos dirigirnos a nuestro perfil y en el tab de Repositorios,

presionamos el botón **New**  y nos muestra un formulario para proceder con la creación del repositorio.



**Create a new repository**

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#). Required fields are marked with an asterisk (\*).

**1 General**

**Owner \***  bvelastegui / **Repository name \***   **practicas\_git is available.**

Great repository names are short and memorable. How about [legendary-octo-lamp](#)?

**Description**

30 / 350 characters

**2 Configuration**

**Choose visibility \***  **Public**

**Start with a template**  **No template**

**Add README**  **Off**

READMEs can be used as longer descriptions. [About READMEs](#)

**Add .gitignore**  **No .gitignore**

.gitignore tells git which files not to track. [About ignoring files](#)

**Add license**  **No license**

Licenses explain how others can use your code. [About licenses](#)

**Connect GitHub Apps**  **Choose apps**

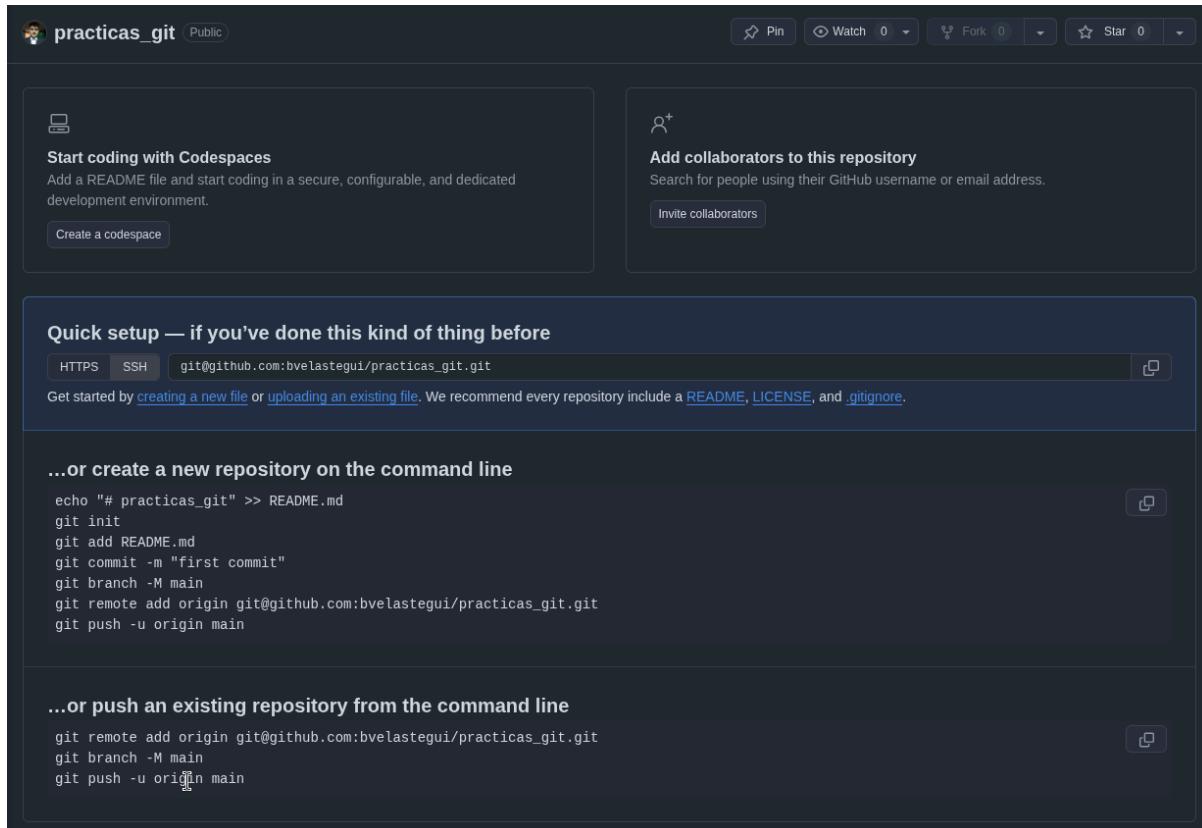
Connect this repository to apps bvelastegui is subscribed to

**Create repository**

Para el nombre del repositorio no podemos usar caracteres especiales y de manera inmediata se nos informará si el nombre que elegimos se encuentra disponible.

En este caso vamos a dejar la visibilidad como pública (*cualquier persona puede verlo y descargarlo*), no agregamos *README* ni *.gitignore* ni *licencia*, solo presionamos el botón *Create repository*.

Si todo está correcto nos redirecciona a la página de nuestro repositorio 



The screenshot shows a GitHub repository page for 'practicas\_git'. At the top, there are buttons for Pin, Watch (0), Fork (0), Star (0), and a dropdown menu. Below the header, there are sections for 'Start coding with Codespaces' (with a 'Create a codespace' button) and 'Add collaborators to this repository' (with a 'Invite collaborators' button). A 'Quick setup — if you've done this kind of thing before' section includes a code editor with 'HTTPS' selected, showing the URL 'git@github.com:bvelastegui/practicas\_git.git'. It also provides instructions for creating a new repository or pushing an existing one from the command line, with examples of git commands.

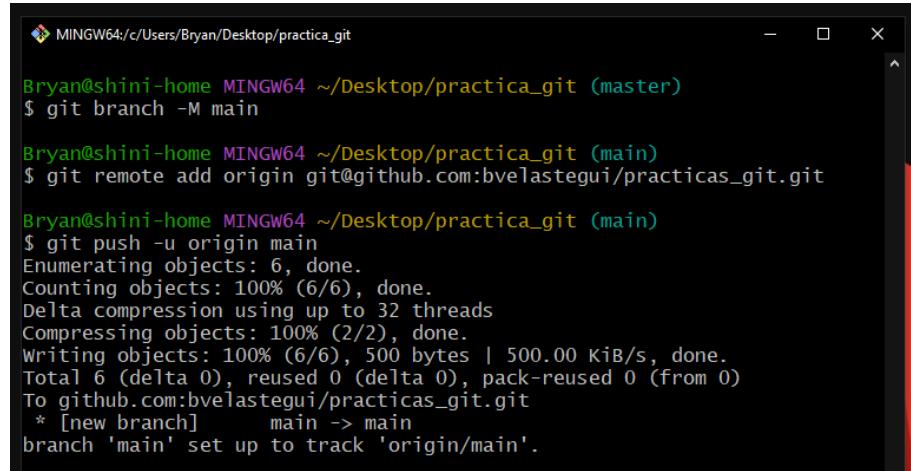
## Subir archivos al repositorio

Para agregar este repositorio como un repositorio remoto en nuestro git vamos a seguir los pasos que se indican en el mismo repositorio de GitHub:

1. `git branch -M main` Crea una nueva rama llamada `main` y le asigna como principal
2. `git remote add origin git@github.com:bvelastegui/practicas\_git.git`

Agrega el repositorio remoto con el nombre de `origin`

3. `git push -u origin main` Empuja/Sube todos los cambios a nuestro repositorio remoto



```

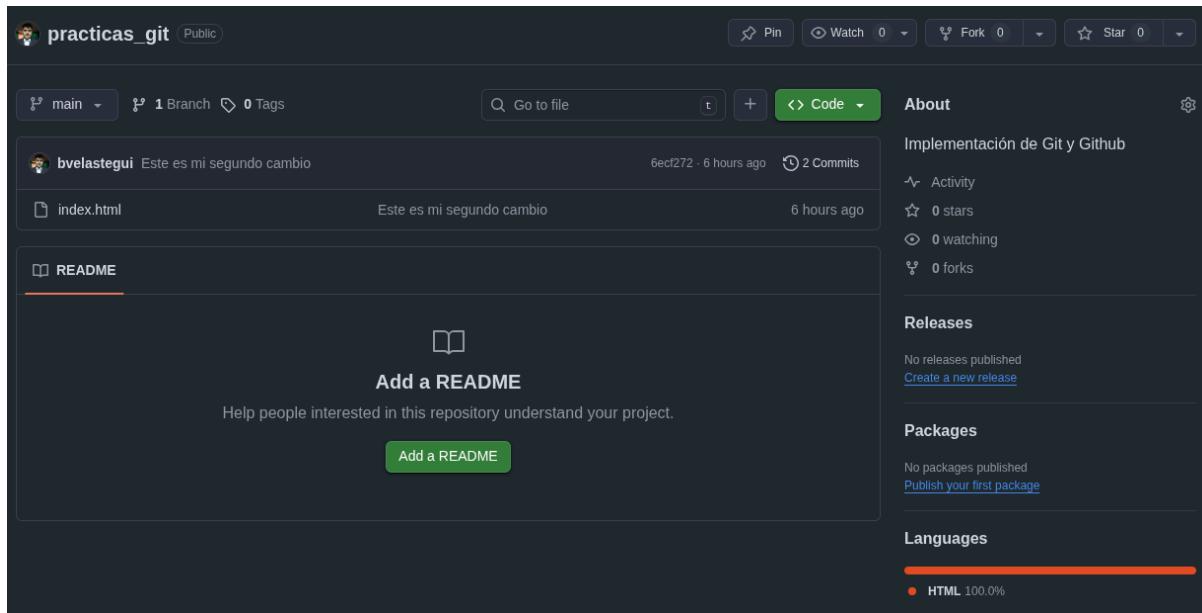
MINGW64:c/Users/Bryan/Desktop/practica_git
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ git branch -M main

Bryan@shini-home MINGW64 ~/Desktop/practica_git (main)
$ git remote add origin git@github.com:bvelastegui/practicas_git.git

Bryan@shini-home MINGW64 ~/Desktop/practica_git (main)
$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 32 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 500 bytes | 500.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:bvelastegui/practicas_git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

```

En github recargamos la página de nuestro repositorio y podemos notar que nuestros archivos se publicaron con éxito.



## Enlaces de repositorios

- Bryan D. Velastegui L. [https://github.com/bvelastegui/web\\_programming\\_cenestur](https://github.com/bvelastegui/web_programming_cenestur)
- Lenin J. Guachamin R.
- Shynlu P. Chimbo T.

## Conclusiones

Git es una herramienta básica en el flujo de trabajo de los desarrolladores, en este laboratorio se pudo aplicar los conocimientos básicos sobre Git y GitHub.

Utilizando Git aprendimos a inicializar un repositorio, agregar archivos para ser observados y confirmar cambios. En GitHub pudimos registrar una clave ssh, crear un repositorio y publicar nuestros archivos locales en un repositorio online (Velastegui, 2025).

## Referencias

- Git (<https://git-scm.com>)
- GitHub (<https://github.com>)
- Visual Studio Code (<https://code.visualstudio.com/>)