

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

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

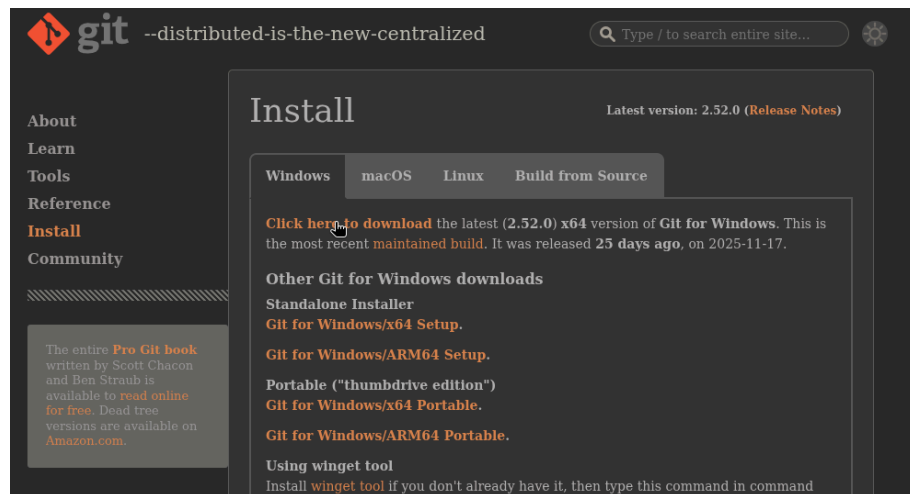
Objetivo

Configurar Git y Github

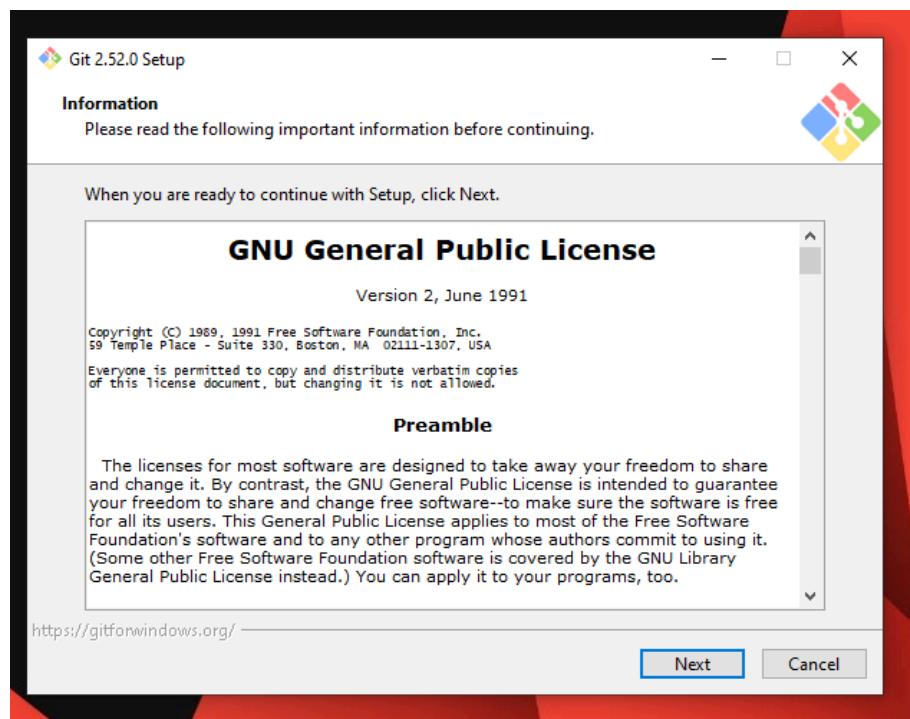
Configuración Git

Instalación

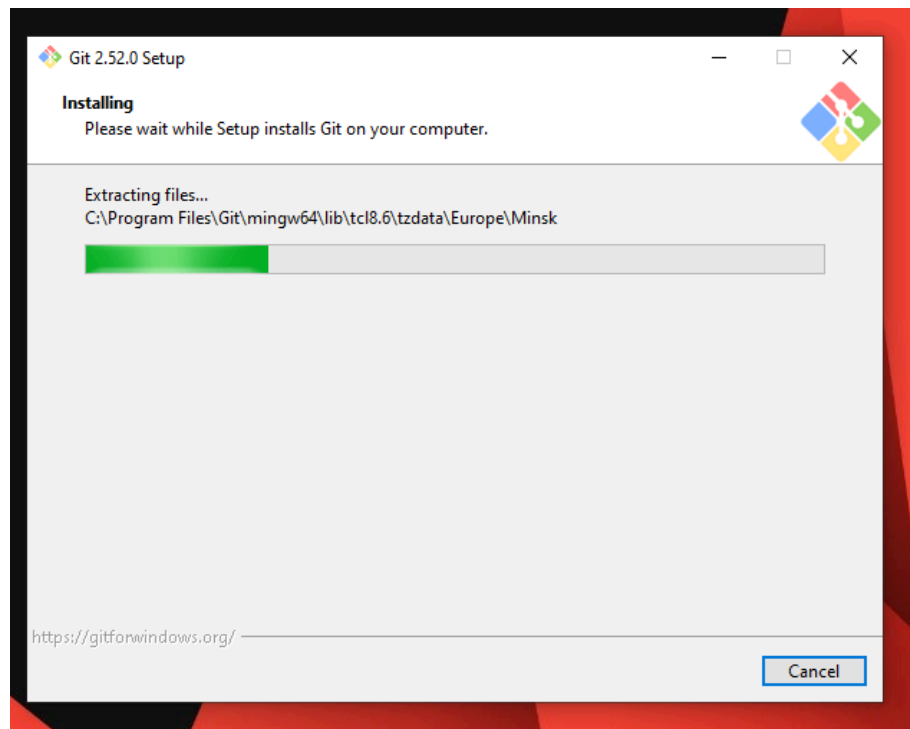
Descargamos el instalador desde la página web de git: <https://git-scm.com/install/windows>



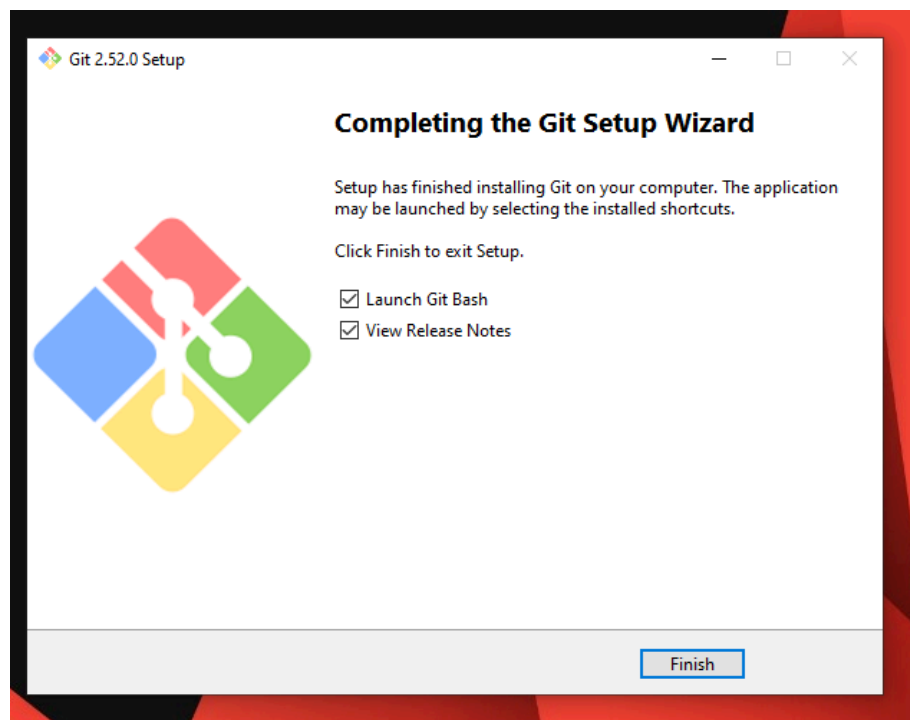
Al completar la descarga ejecutamos el instalador



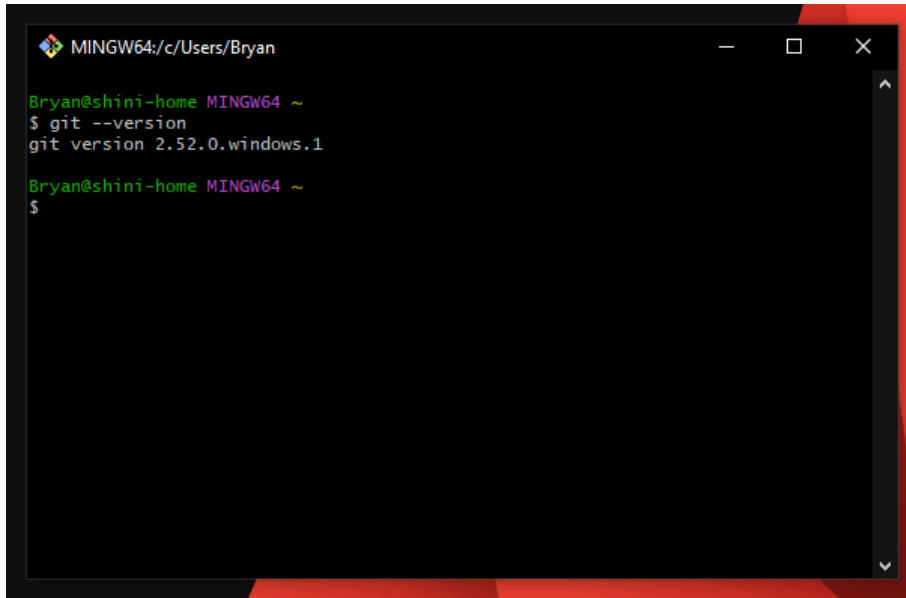
Utilizamos todas configuraciones por defecto y procedemos con la instalación



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.

A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/Bryan'. The prompt is 'Bryan@shini-home MINGW64 ~'. The user enters '\$ git --version' and the output is 'git version 2.52.0.windows.1'. The prompt returns to '\$'.

```
MINGW64:/c/Users/Bryan

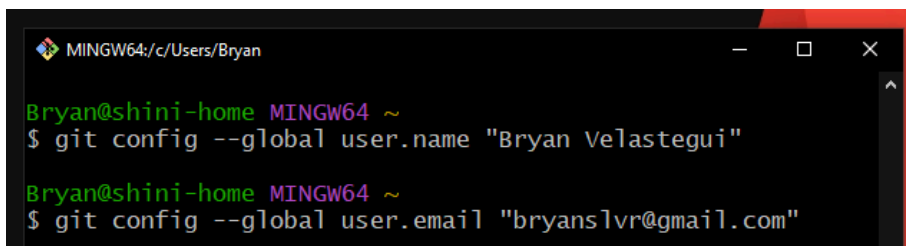
Bryan@shini-home MINGW64 ~
$ git --version
git version 2.52.0.windows.1
Bryan@shini-home MINGW64 ~
$
```

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'
```

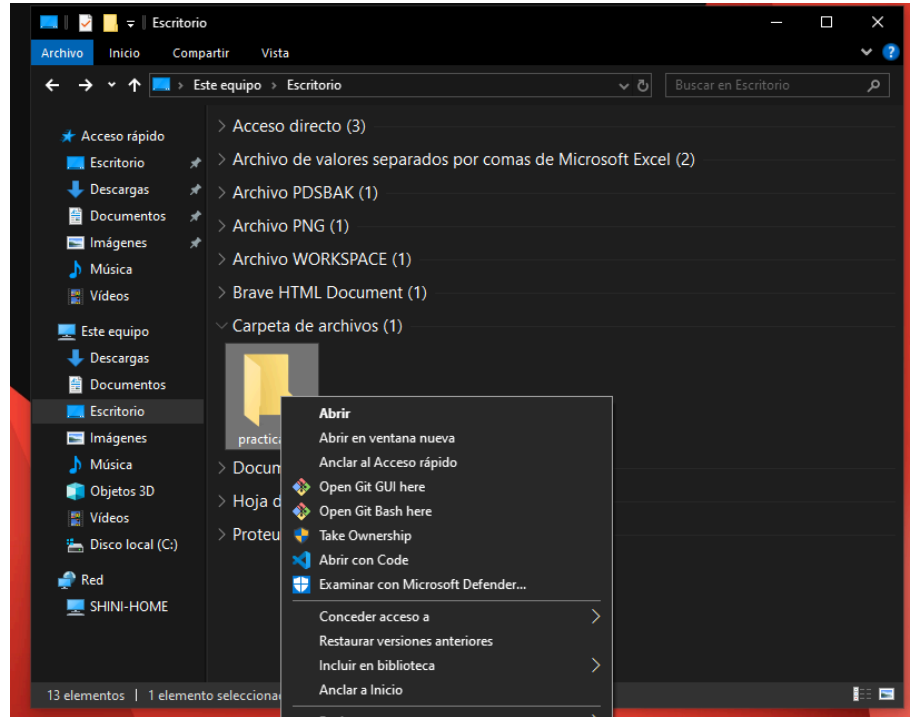
A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/Bryan'. The prompt is 'Bryan@shini-home MINGW64 ~'. The user enters '\$ git config --global user.name "Bryan Velastegui"' and the prompt returns to '\$'. Then the user enters '\$ git config --global user.email "bryanslvr@gmail.com"' and the prompt returns to '\$'.

```
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"
$
```

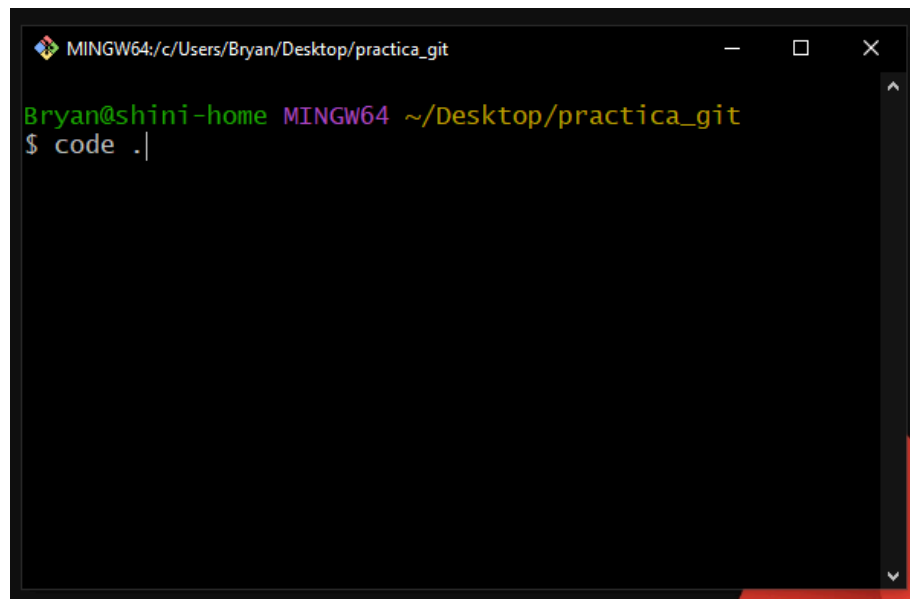
Práctica

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`

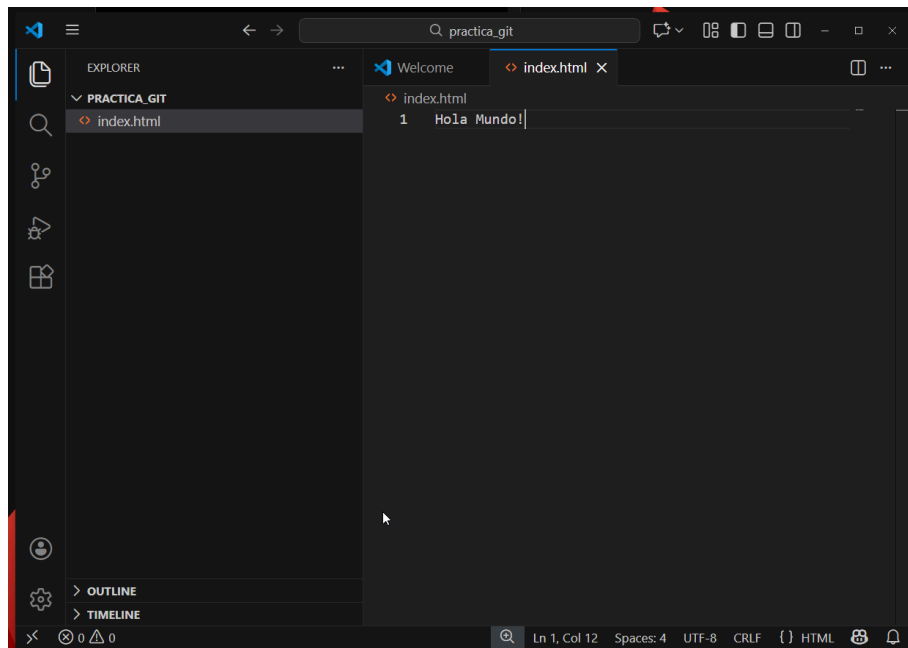


En la consola usamos el comando `code .` para abrir nuestra carpeta usando Visual Studio Code:

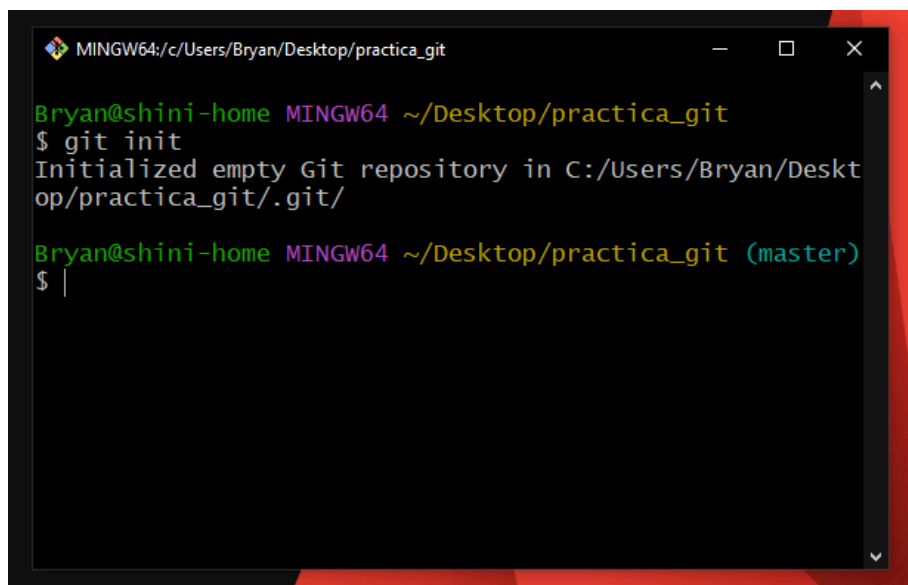
Code:



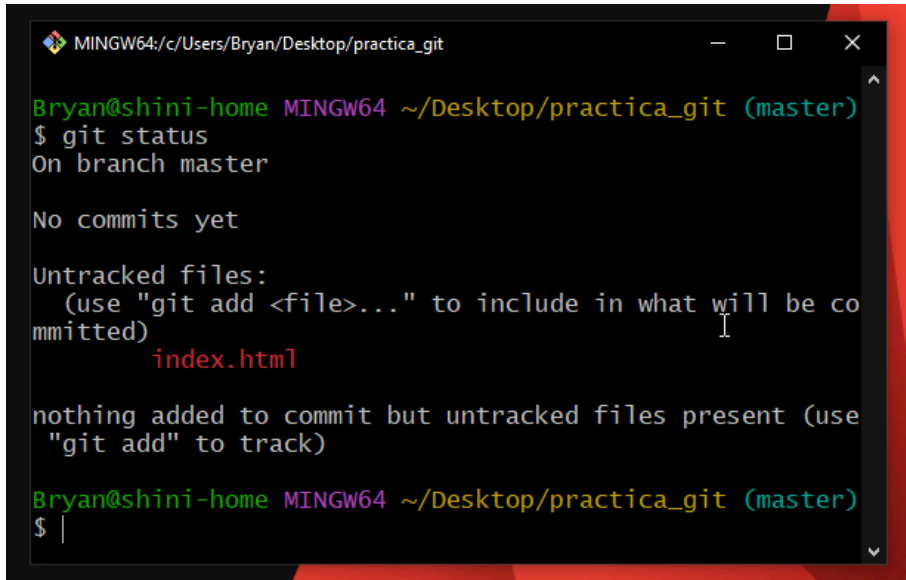
En vs code creamos un nuevo archivo llamado index.html y agregamos hola mundo en su contenido.



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



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

A terminal window titled 'MINGW64:/c/Users/Bryan/Desktop/practica_git' showing the output of the 'git status' command. The output indicates that there are no commits yet and that 'index.html' is an untracked file. It suggests using 'git add' to track the file.

```
MINGW64:/c/Users/Bryan/Desktop/practica_git

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

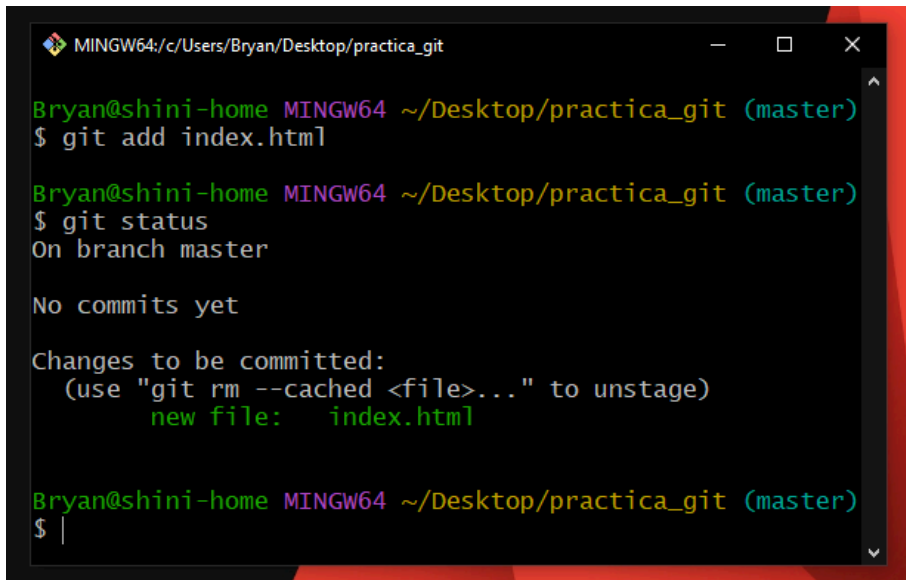
No commits yet

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)

A terminal window titled 'MINGW64:/c/Users/Bryan/Desktop/practica_git' showing the output of 'git add index.html' followed by 'git status'. The output shows that 'index.html' is now a new file staged for commit.

```
MINGW64:/c/Users/Bryan/Desktop/practica_git

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

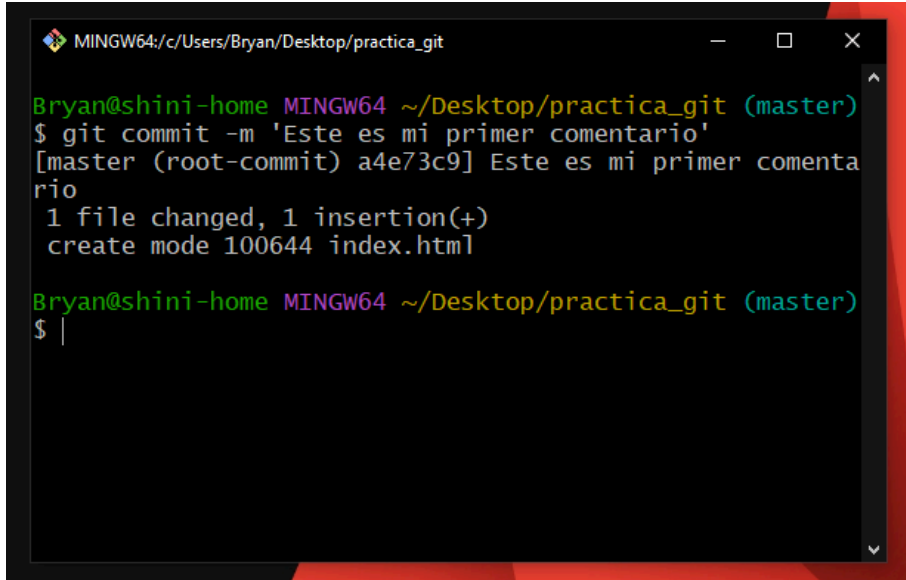
No commits yet

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

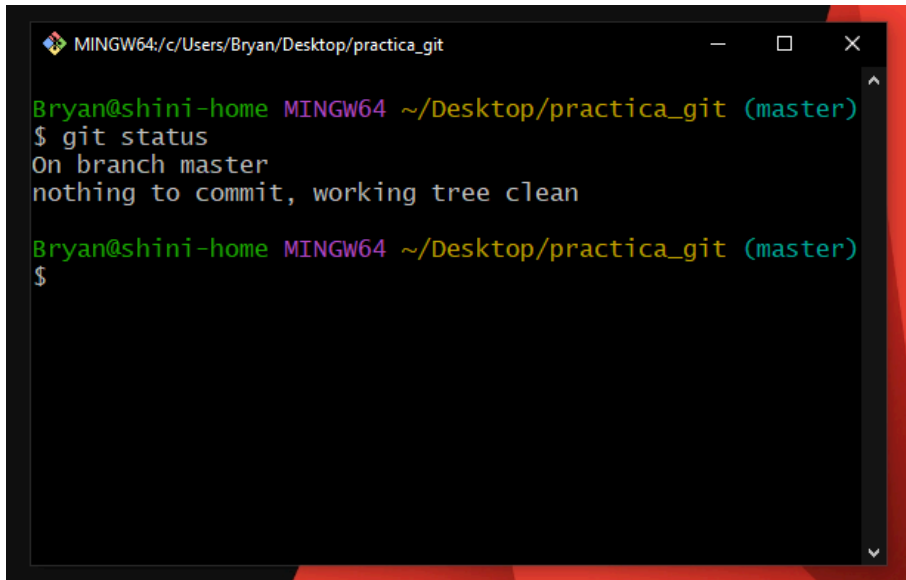
A terminal window titled 'MINGW64: c:/Users/Bryan/Desktop/practica_git' showing a successful git commit. The user 'Bryan@shini-home' is in the 'master' branch. The command executed is 'git commit -m 'Este es mi primer comentario''. The output shows the commit hash 'a4e73c9' and a summary of changes: '1 file changed, 1 insertion(+), create mode 100644 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.

A terminal window titled 'MINGW64: c:/Users/Bryan/Desktop/practica_git' showing the output of the 'git status' command. The user 'Bryan@shini-home' is in the 'master' branch. The command executed is 'git status'. The output indicates the working tree is clean: 'On branch master, nothing to commit, working tree clean'.

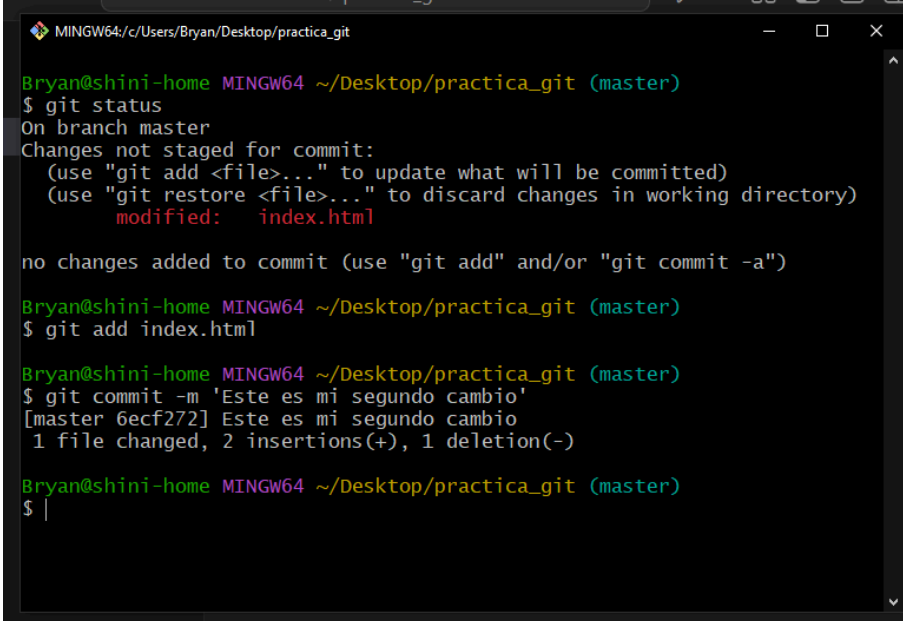
```
MINGW64: c:/Users/Bryan/Desktop/practica_git

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](#).

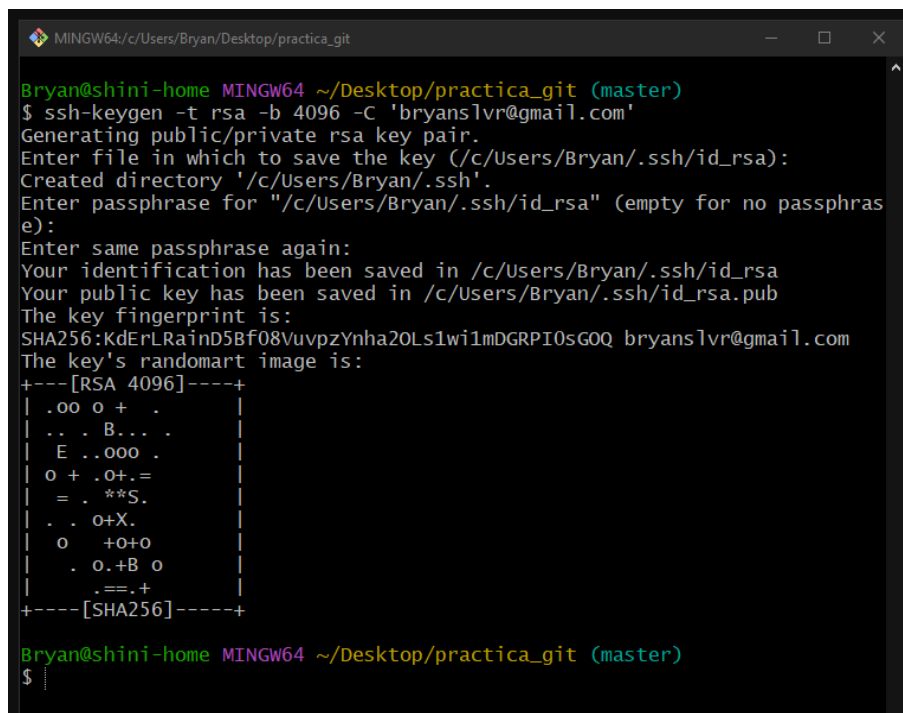
Creación y configuración de clave ssh

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:

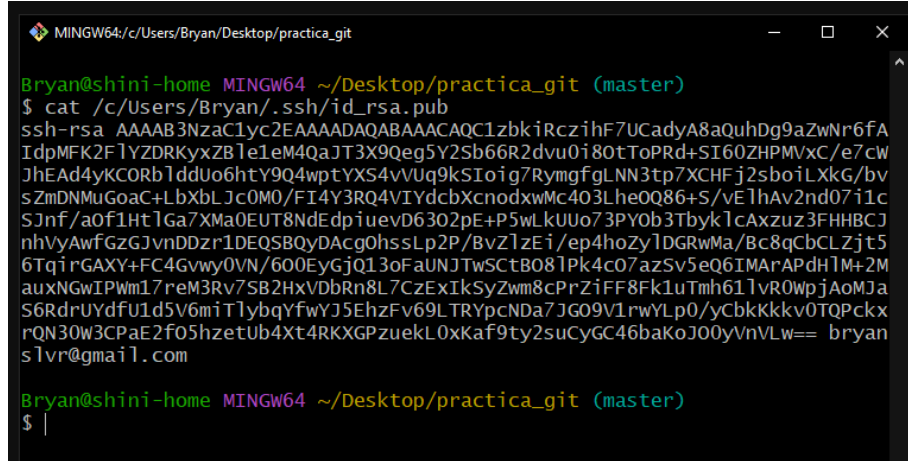
```
ssh-keygen -t rsa -b 4096 -C 'bryanslvr@gmail.com'
```

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')



```
MINGW64/c/Users/Bryan/Desktop/practica_git
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:KdErLRainD5Bf08VuvpzYnha20Ls1wi1mDGRPI0sGOQ bryanslvr@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
| .00 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:



```

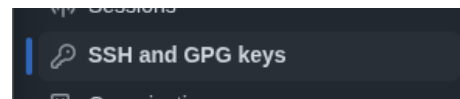
MINGW64/c/Users/Bryan/Desktop/practica_git
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ cat /c/Users/Bryan/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC1zbkiRczihF7UCadyA8aQuhDg9aZwNr6fA
IdpMFK2F1YZDRKyxBle1eM4QaJT3X9Qeg5Y2Sb66R2dvu0i8OtToPRd+SI60ZHPMVxc/e7cW
JhEAd4yKCORblddUo6htY9Q4wptYXS4vVUq9kSIoig7RymgfgLNN3tp7XCHFj2sboiLXkG/bv
sZmDNMuGoaC+LbXbLJc0M0/FI4Y3RQ4VIYdcbXcnodxwMc4O3LheOQ86+S/vE1hAv2nd07i1c
SJnf/aOf1Ht1Ga7XMa0EUT8NdEdpiuevD63O2pE+P5wLkUuo73PYOb3Tbyk1cAxzuz3FHHBCJ
nhVyAwfGzGJvnDDzr1DEQSBQyDacgOhssLp2P/BvZlzei/ep4hoZy1DGRwMa/Bc8qCbCLZjt5
6TqirGAXY+FC4Gvwy0VN/600EyGjQ13oFaUNJTWSCTB081Pk4c07azSv5eQ6IMArAPdHlM+2M
auxNGWIPWm17reM3Rv7SB2HxVDbRn8L7CzExIkSyZwm8cPrZiFF8Fk1uTmh611vR0WpjAoMJa
S6RdrUYdfU1d5V6miTlybqYfwYJ5EhzFv69LTRYpcNda7JG09V1rwYLP0/yCbkkkkv0TQPckx
rQN30W3CPaE2f05hzetUb4Xt4RKXGPzuekL0xKaf9ty2suCyGC46baKoJ00yVnVLw== bryan
slvr@gmail.com

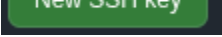
Bryan@shini-home MINGW64 ~/Desktop/practica_git (master)
$ |

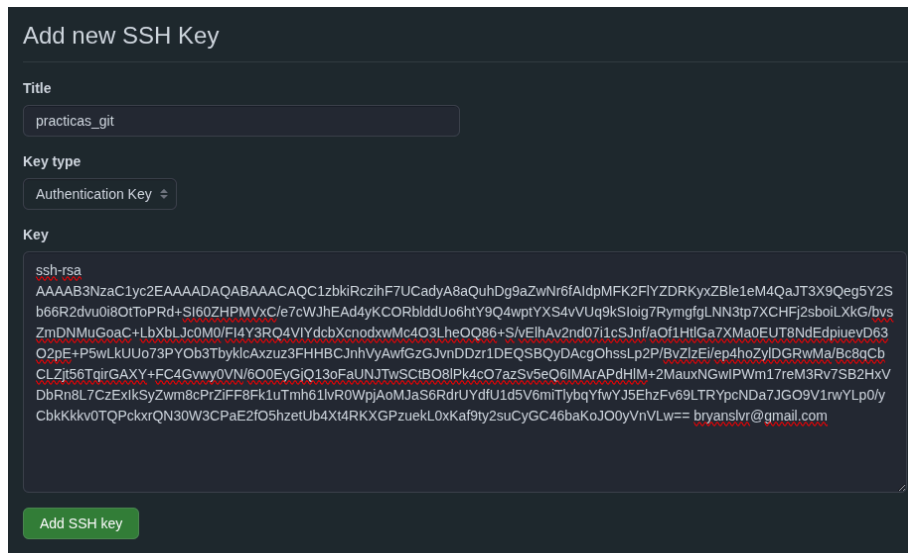
```

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 AAAAB3NzaC1yc2EAAAADAQABAAQCAQC1zbkiRczihF7UCadyA8aQuhDg9aZwNr6fAIdpMFK2FIYZDRKyxBle1eM4QaJT3X9Qeg5Y2Sb66R2dvu0i8OtToPRd+SI60ZHPMVxc/e7cWJhEAd4yKCORblddUo6htY9Q4wptYXS4vVUq9kSIoig7RymgfgLNN3tp7XCHFj2sboiLXkG/bvsZmDNMuGoaC+LbXbLJc0M0/FI4Y3RQ4VIYdcbXcnodxwMc4O3LheOQ86+S/vE1hAv2nd07i1cSJnf/aOf1Ht1Ga7XMa0EUT8NdEdpiuevD63O2pE+P5wLkUuo73PYOb3Tbyk1cAxzuz3FHHBCJnhVyAwfGzGJvnDDzr1DEQSBQyDacgOhssLp2P/BvZlzei/ep4hoZy1DGRwMa/Bc8qCbCLZjt56TqirGAXY+FC4Gvwy0VN/600EyGjQ13oFaUNJTWSCTB081Pk4c07azSv5eQ6IMArAPdHlM+2MauxNGWIPWm17reM3Rv7SB2HxVDbRn8L7CzExIkSyZwm8cPrZiFF8Fk1uTmh611vR0WpjAoMJaS6RdrUYdfU1d5V6miTlybqYfwYJ5EhzFv69LTRYpcNda7JG09V1rwYLP0/yCbkkkkv0TQPckxrQN30W3CPaE2f05hzetUb4Xt4RKXGPzuekL0xKaf9ty2suCyGC46baKoJ00yVnVLw== bryanslvr@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*)

SSH keys

[New SSH key](#)


This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

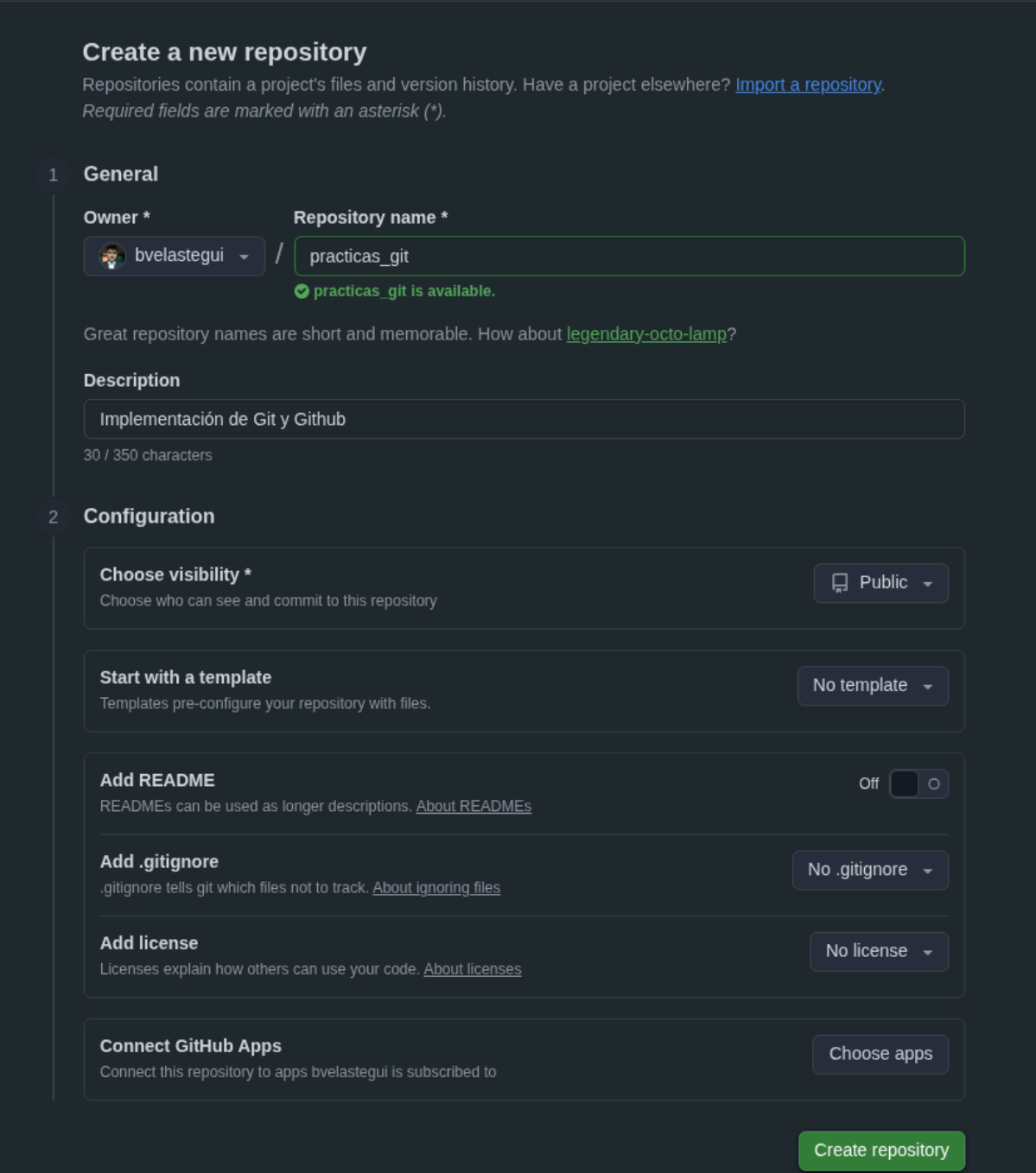
Authentication keys

 SSH	Ubuntu 22.04 LTS SHA256:YHeIydbdPUPhw9L/UDL5BP5YMIRT5svX2GaEc1VQ7fE Added on Sep 16, 2023 Last used within the last week — Read/write	Delete
 SSH	practicas_git SHA256:KdErLRa1nD5Bf08VuvpzYnha20Ls1w11mDGRPI0sG0Q Added on Dec 12, 2025 Never used — Read/write	Delete

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 * **Repository name ***

 bvelastegui / practicas_git

✔ practicas_git is available.

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

Description

Implementación de Git y Github

30 / 350 characters

2 **Configuration**

Choose visibility * Choose who can see and commit to this repository Public

Start with a template Templates pre-configure your repository with files. No template

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

Add .gitignore .gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Add license Licenses explain how others can use your code. [About licenses](#) No license

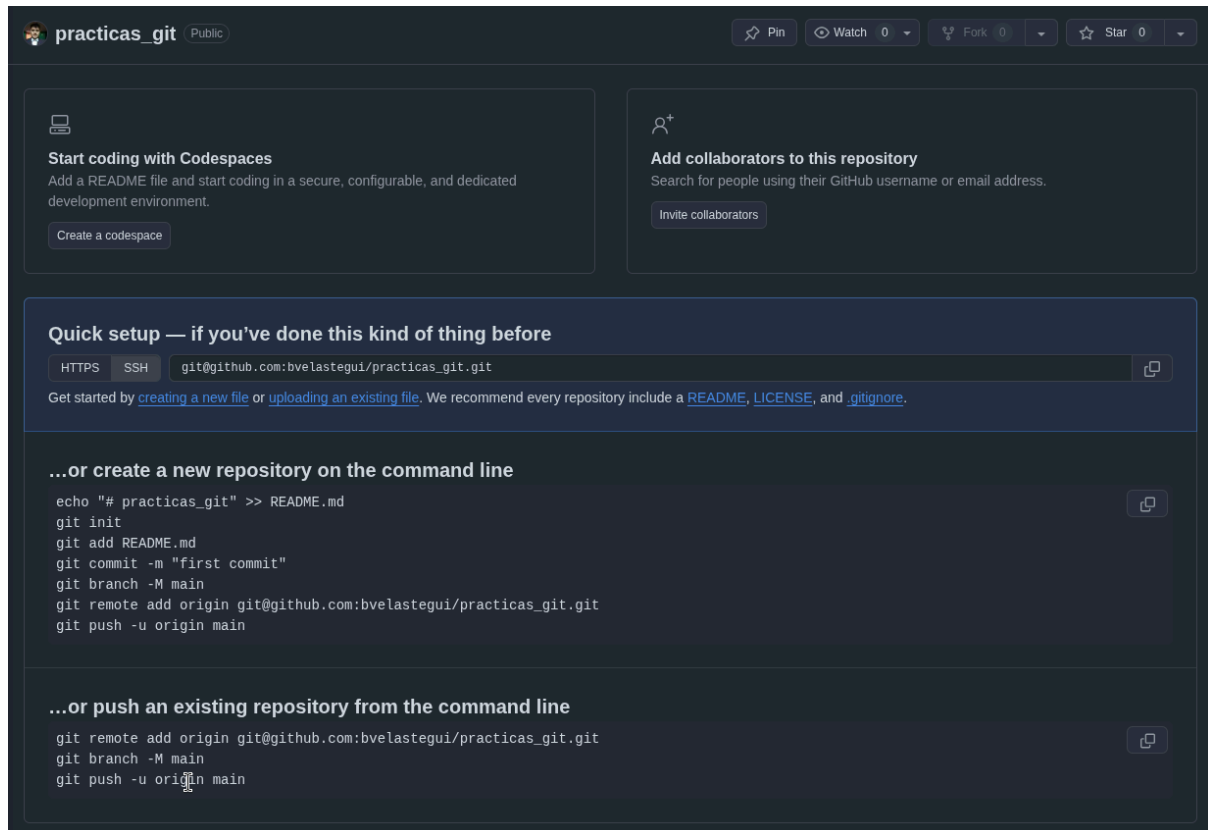
Connect GitHub Apps Connect this repository to apps bvelastegui is subscribed to Choose apps

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 🎉



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:

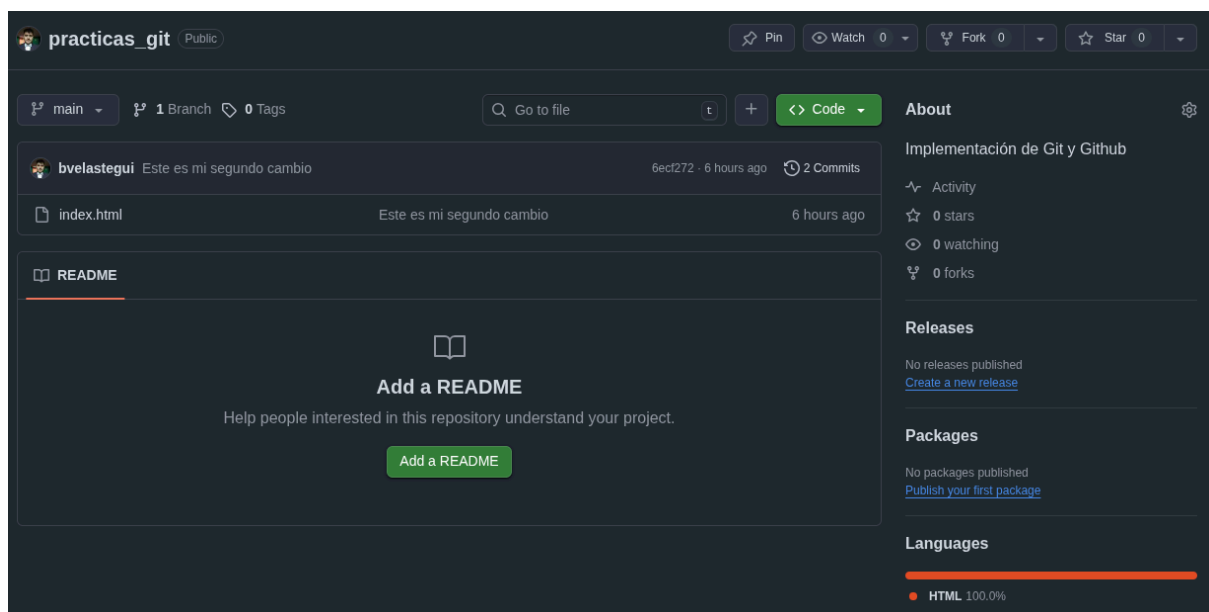
1. `git branch -M main` Crea una nueva rama llamada main y le asigna como principal
2. `git remote add origin git@github.com:bvelastegui/practicass_git.git`

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
- 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 (Velasategui, 2025).

Referencias

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