

“INSTALACIÓN Y CONFIGURACIÓN DE HERRAMIENTA DE VERSIONAMIENTO  
(LOCAL / WEB) ”

PRESENTADO A:  
CRISTIAN CAMILO ARIAS

PRESENTADO POR:  
ANGIE VANESSA PALACIO VILLEGAS



## INTRODUCCIÓN

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git.

Se utiliza principalmente para la creación de código fuente de programas de ordenador.

En este documento validamos y realizaremos la instalación, configuración del Git local y git remoto.

## OBJETIVO

Se realizará la instalación y configuración correcta de GIT.

Git local

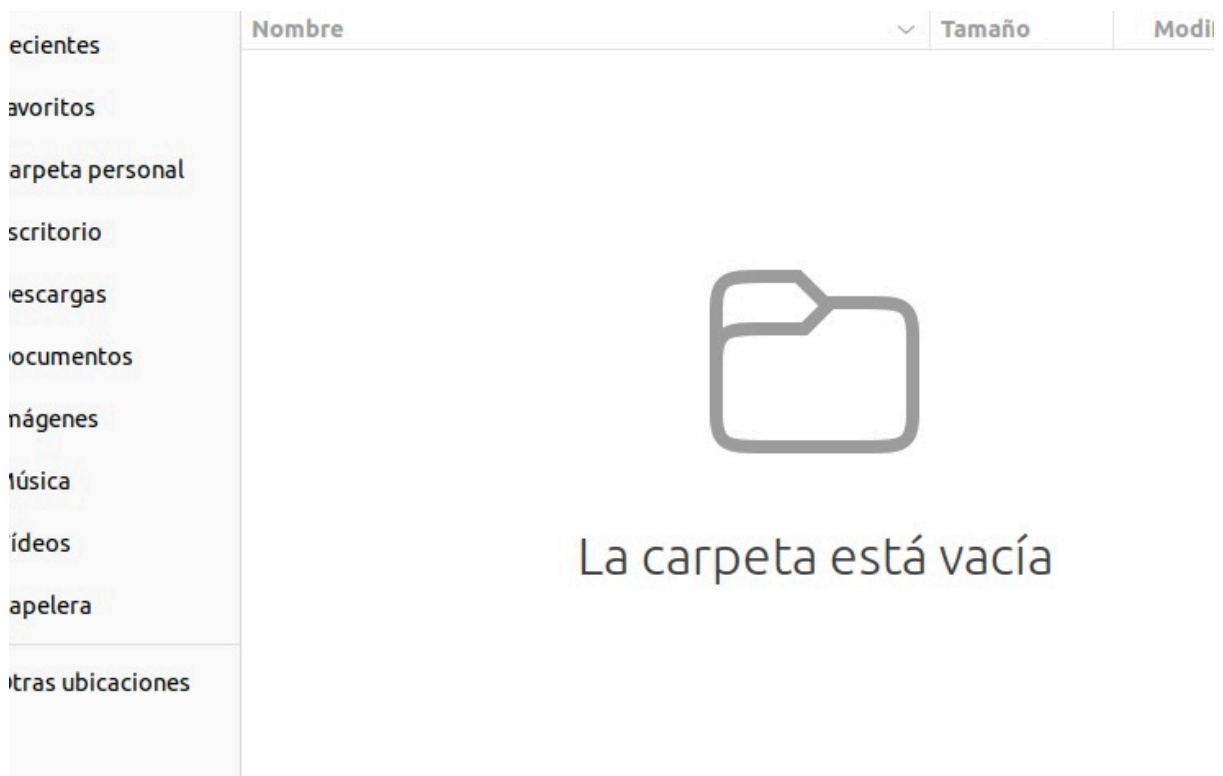
Se realiza la instalación, en mi caso ya se encontraba instalado en mi equipo

```
angie@angie-X541NA:~$ sudo apt-get install git
[sudo] contraseña para angie:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
git ya está en su versión más reciente (1:2.34.1-1ubuntu1.10).
fijado git como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 153 no actualizados.
```

Se crea carpeta en Escritorio con el nombre de taller, se llama a la consola

```
angie@angie-X541NA: ~/Escritorio/taller
angie@angie-X541NA:~$ cd Escritorio/taller/
angie@angie-X541NA:~/Escritorio/taller$
```

Se realiza la validación carpeta esta sola

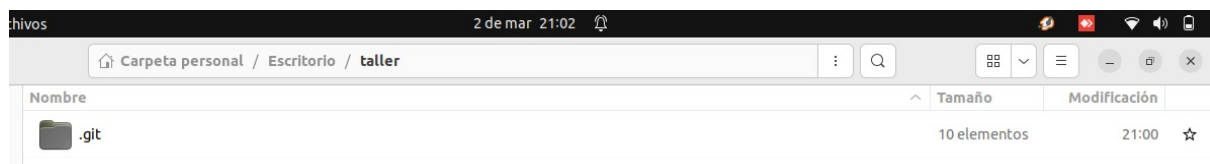




Se crea el Git en la carpeta con el comando

```
angie@angie-X541NA:~/Escritorio/taller$ git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda:
ayuda: git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
ayuda:
ayuda: git branch -m <nombre>
Iniciado repositorio Git vacío en /home/angie/Escritorio/taller/.git/
angie@angie-X541NA:~/Escritorio/taller$
```

Se crea la carpeta GIT



Copiamos dos archivos para tener en la carpeta y brindamos un git status

```
angie@angie-X541NA:~/Escritorio/taller$ git status .
En la rama master

No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
  Ejercicio1.html
  Taller frontend/

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
angie@angie-X541NA:~/Escritorio/taller$
```

despues agregamos los dos archivos con un git add y brindamos un git status

```
angie@angie-X541NA:~/Escritorio/taller$ git add .
angie@angie-X541NA:~/Escritorio/taller$ git status .
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)
  nuevos archivos: Ejercicio1.html
  nuevos archivos: Taller_frontend/CSS/style.css
  nuevos archivos: Taller_frontend/Taller frontend.html
  nuevos archivos: Taller_frontend/index.html
  nuevos archivos: Taller_frontend/js/index.js

angie@angie-X541NA:~/Escritorio/taller$
```

continuamos con un git commit



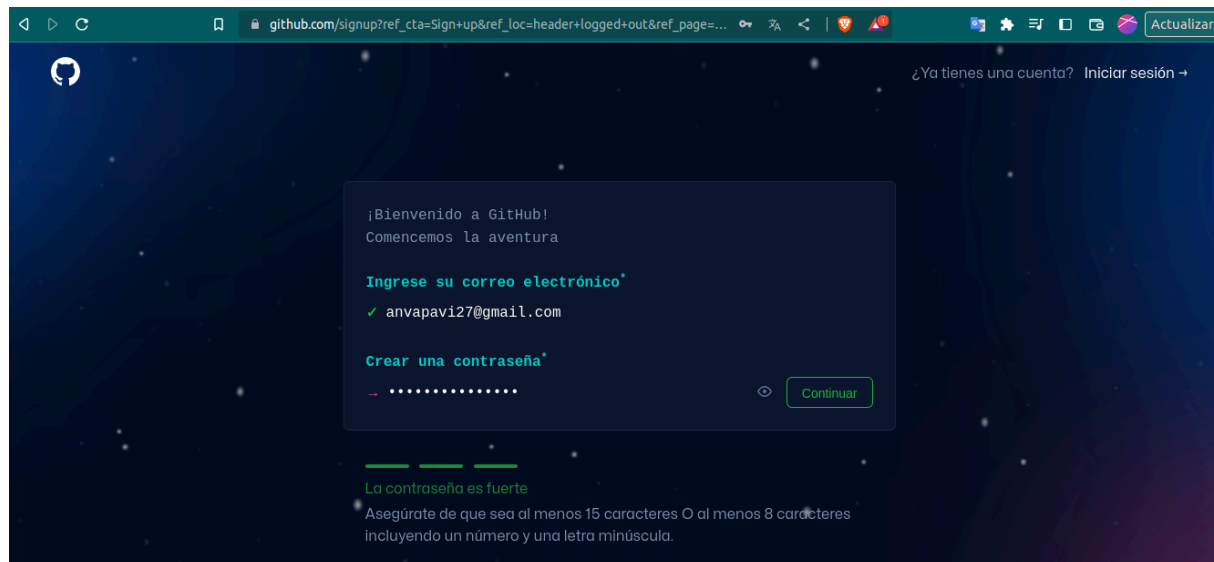
```
angie@angie-X541NA:~/Escritorio/taller$ git commit -m 'taller git'
[master (commit-raiz) 5383b33] taller git
5 files changed, 163 insertions(+)
create mode 100644 Ejercicio1.html
create mode 100644 Taller_frontend/CSS/style.css
create mode 100644 Taller_frontend/Taller_froond.html
create mode 100644 Taller_frontend/index.html
create mode 100644 Taller_frontend/js/index.js
angie@angie-X541NA:~/Escritorio/taller$
```

y brindamos nuevamente un git status

```
angie@angie-X541NA:~/Escritorio/taller$ git status
En la rama master
nada para hacer commit, el árbol de trabajo está limpio
```

## Git remoto

Primero debemos de registrarnos



Después de registrarnos vamos a crear un repositorio para subir nuestras carpetas



**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*  / Repository name \*

Ads is available

Great repository names are short and memorable. Need inspiration? How about [stunning-pancake](#) ?

Description (optional)

☐ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

☐ **Add .gitignore**

Despues de crearlo vamos a seguir el paso a paso de los comandos en nuestra consola vamos abrir uan nueva carpeta que se va llamar taller 2 y iniciamos con el primer comando

**Set up GitHub Copilot**

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

[Get started with GitHub Copilot](#)

**Quick setup — if you've done this kind of thing before**

[HTTPS](#) [SSH](#) <https://github.com/apalacio09/Adso.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository has a README file.

**...or create a new repository on the command line**

```
echo "# Adso" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/apalacio09/Adso.git
git push -u origin main
```

```
angie@angie-X541NA: ~/Escritorio/taller2
angie@angie-X541NA:~/Escritorio/taller2$ git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama pr
edeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para
usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda: git config --global init.defaultBranch <nombre>
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk'
y
ayuda: 'development'. Se puede cambiar el nombre de la rama creada median
te este comando:
ayuda: git branch -M <nombre>
Inicializado repositorio Git vacio en /home/angie/Escritorio/taller2/.git/
angie@angie-X541NA:~/Escritorio/taller2$
```

Despues seguimos con el segundo comando



```
angie@angie-X541NA:~/Escritorio/taller2$ git commit -m "first commit"
En la rama master

Confirmación inicial

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
    Ejercicio1.html
    Taller_frontend/

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa
"git add" para hacerles seguimiento)
angie@angie-X541NA:~/Escritorio/taller2$
```

Vamos a brindar un git status

```
angie@angie-X541NA:~/Escritorio/taller2$ git status
En la rama master

No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
    Ejercicio1.html
    Taller_frontend/

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa
"git add" para hacerles seguimiento)
angie@angie-X541NA:~/Escritorio/taller2$
```

Vamos a agregar las carpetas

```
angie@angie-X541NA:~/Escritorio/adso$ git add .
angie@angie-X541NA:~/Escritorio/adso$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Cambios a ser confirmados:
  (usa "git restore --staged <archivo>..." para sacar del área de stage)
    nuevos archivos: Ejercicio1.html
    nuevos archivos: Taller_frontend/CSS/style.css
    nuevos archivos: Taller_frontend/Taller froond.html
    nuevos archivos: Taller_frontend/index.html
    nuevos archivos: Taller_frontend/js/index.js
```

Vamos a realizar un commit

```
angie@angie-X541NA:~/Escritorio/adso$ git commit -m "trabajo git"
[main e454f3c] trabajo git
5 files changed, 163 insertions(+)
create mode 100644 Ejercicio1.html
create mode 100644 Taller_frontend/CSS/style.css
create mode 100644 Taller_frontend/Taller froond.html
create mode 100644 Taller_frontend/index.html
create mode 100644 Taller_frontend/js/index.js
```



## llamamos al remoto

```
no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa 'git add' para hacerlos seguimiento).
angie@angie-X541NA:~/Escritorio/taller2$ git branch -M main
angie@angie-X541NA:~/Escritorio/taller2$ git remote add origin https://github.com/apalacio09/Adso.git
angie@angie-X541NA:~/Escritorio/taller2$
```

## Subimos los archivos

```
angie@angie-X541NA:~/Escritorio/adso$ git push -u origin main
Username for 'https://github.com': apalacio09
Password for 'https://apalacio09@github.com':
Enumerando objetos: 11, listo.
Contando objetos: 100% (11/11), listo.
Compresión delta usando hasta 2 hilos
Comprimiendo objetos: 100% (8/8), listo.
Escribiendo objetos: 100% (10/10), 2.73 KiB | 932.00 KiB/s, listo.
Total 10 (delta 0), reusados 0 (delta 0), pack-reusados 0
To https://github.com/apalacio09/Adso.git
    e091740..e454f3c  main -> main
Rama 'main' configurada para hacer seguimiento a la rama remota 'main' de 'origin'.
```

The screenshot shows the GitHub interface for a repository named 'Adso' by user 'apalacio09'. The repository is public and has 1 branch (main) and 0 tags. The file list shows 'Taller\_frontend', 'Ejercicio1.html', and 'README.md'. The README content is visible, showing the title 'Adso'. The right sidebar contains sections for 'About', 'Releases', and 'Packages', all indicating no content has been published yet.

File	Commit	Time
Taller_frontend	trabajo git	1 hour ago
Ejercicio1.html	trabajo git	1 hour ago
README.md	first commit	1 hour ago