

# GIT Y GITHUB

---



HOLA, MI NOMBRE ES VIKI, TE INVITO A QUE APRENDAMOS JUNTOS ACERCA DEL CONTROL DE VERSIONES, ES UN TEMA QUE DE SEGURO TE FACILITARÁ EL TRABAJO COLABORATIVO



# MODULOS

MODULO 1

INTRO

MODULO 2

INSTALACIÓN

MODULO 3

SHE INVITACIÓN

MODULO 4

COMMITs

MODULO 5

RAMAS

# MODULO 1

# INTRO

# UN POCO DE TEORIA

## Acerca de los repositorios

Un repositorio contiene todos los archivos de tu proyecto y el historial de revisiones de cada uno de ellos. Puedes debatir y administrar el trabajo de tu proyecto dentro del repositorio.

Puedes ser propietario de repositorios individualmente o puedes compartir la propiedad de los repositorios con otras personas en una organización.

Puedes restringir quién tiene acceso a un repositorio seleccionando la visibilidad del mismo. Para obtener más información, consulta la sección "Acerca de la visibilidad de un repositorio".



Puedes utilizar repositorios para administrar tu trabajo y colaborar con otros.

Puedes utilizar propuestas para recolectar la retroalimentación de los usuarios, reportar errores de software y organizar las tareas que te gustaría realizar.

Puedes utilizar los GitHub Discussions para hacer y responder preguntas, compartir información, hacer anuncios y conducir o participar en las conversaciones sobre un proyecto.

Puedes utilizar las solicitudes de cambios para proponer cambios a un repositorio.

Puedes utilizar tableros de proyecto para organizar y priorizar tus propuestas y solicitudes de cambios.

Los repositorios y archivos individuales están sujetos a límites de tamaño.



# ELABORADO POR LOS MEJORES

DIEGO ALZATE

ALEJANDRO GIRALDO

JACOB HOYOS

DIANA RESTREPO

JULIAN FRANCO



# GIT Y GITHUB

---



HOLA, EN ESTE MODULO  
APRENDEREMOS COMO ADQUIRIR  
E INSTALAR LAS HERRAMIENTAS  
GIT Y GITHUB



# MODULO 2

# INSTALACIÓN

# CÓMO ADQUIRIR GIT



Para la obtención del creador de repositorios Git, lo consigues a través de la página [www.git-scm.com/download](http://www.git-scm.com/download). Seleccionamos el Sistema Operativo a trabajar y luego damos clic en descargar para windows o si eres linux copiaras la linea de código

 **git** --local-branching-on-the-cheap

Search entire site...

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), convenient staging areas, and [multiple workflows](#).



**About**  
The advantages of Git compared to other source control systems.

**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases for all major platforms.

**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.

 **Pro Git** by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

**Latest source Release**  
**2.35.1**  
[Release Notes \(2022-01-29\)](#)

[Download for Windows](#)



[Windows GUIs](#)   [Tarballs](#)  
[Mac Build](#)   [Source Code](#)

**Companies & Projects Using Git**

Google FACEBOOK Microsoft   NETFLIX  PostgreSQL   
      

# CÓMO ADQUIRIR GIT



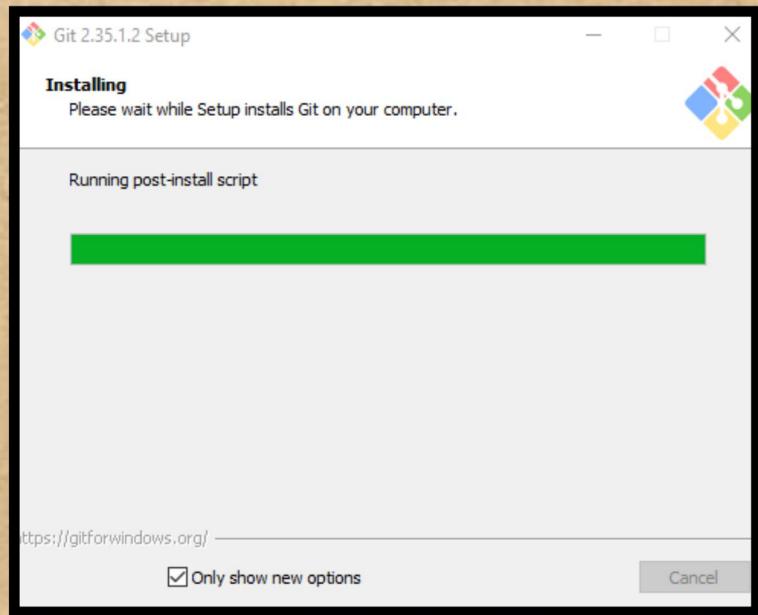
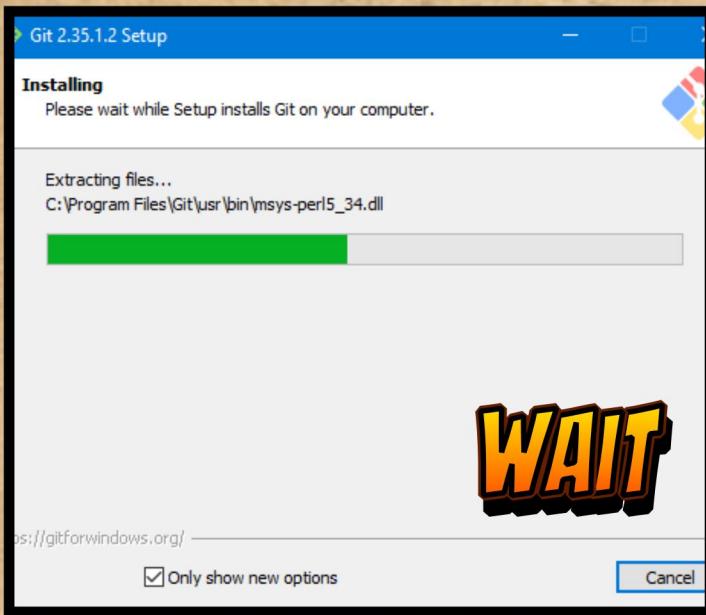
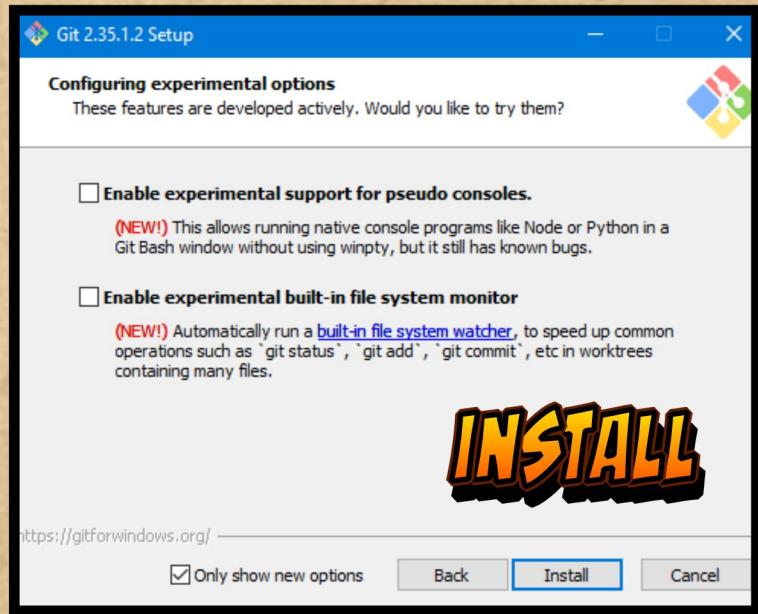
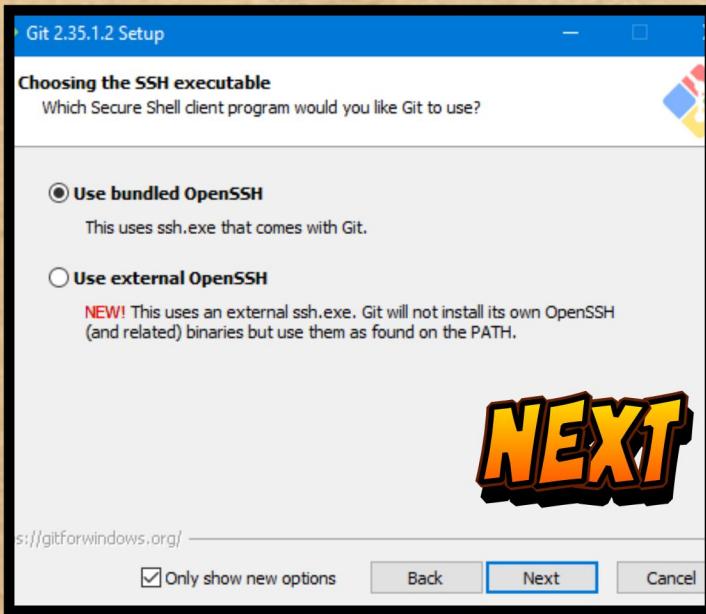
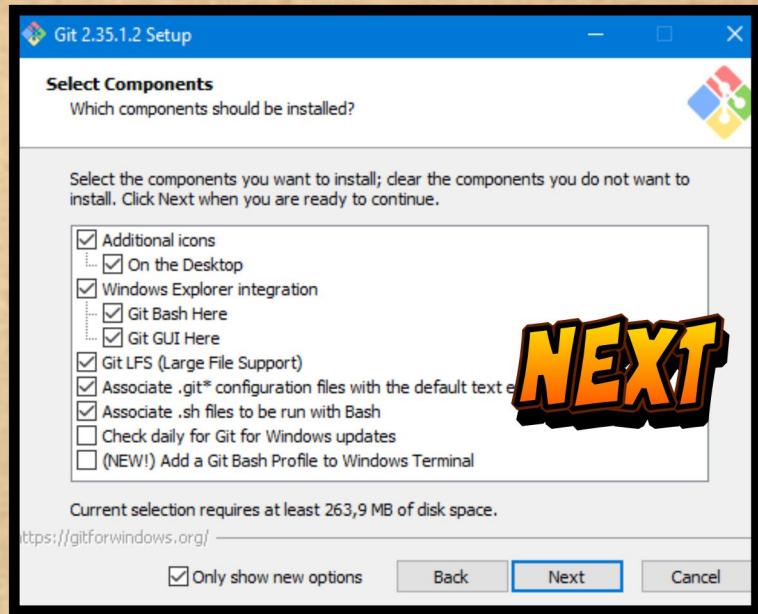
Empezaremos desde el principio, en este caso instalando la herramienta GIT y luego implementamos la conexión a GITHUB.

The screenshot shows the official Git website at [git-scm.com](https://git-scm.com). The main navigation menu includes 'About', 'Documentation', 'Downloads' (selected), 'Community', and 'Book'. Under 'Downloads', links are provided for 'GUI Clients', 'Logos', and 'Community'. A sidebar on the left contains information about the 'Pro Git book'. The main content area is titled 'Download for Windows' and provides links for 'Windows Downloads', 'Standalone Installer', '32-bit Git for Windows Setup', '64-bit Git for Windows Setup', 'Portable ("thumbdrive edition")', '32-bit Git for Windows Portable', and '64-bit Git for Windows Portable'. It also includes instructions for using the 'winget tool' and a command-line example: `winget install --id Git.Git -e --source winget`.

The screenshot shows the official Git website at [git-scm.com](https://git-scm.com). The main navigation menu includes 'About', 'Documentation', 'Downloads' (selected), 'Community', and 'Book'. Under 'Downloads', links are provided for 'GUI Clients', 'Logos', and 'Community'. The main content area is titled 'Download for Linux and Unix' and provides instructions for installing Git on Linux using the package manager of your distribution. It lists 'Debian/Ubuntu', 'Fedora', 'Gentoo', 'Arch Linux', and 'openSUSE' with their respective installation commands. A note states: 'It is easiest to install Git on Linux using the preferred package manager of your Linux distribution. If you prefer to build from source, you can find tarballs on [kernel.org](http://kernel.org). The latest version is 2.35.1.'

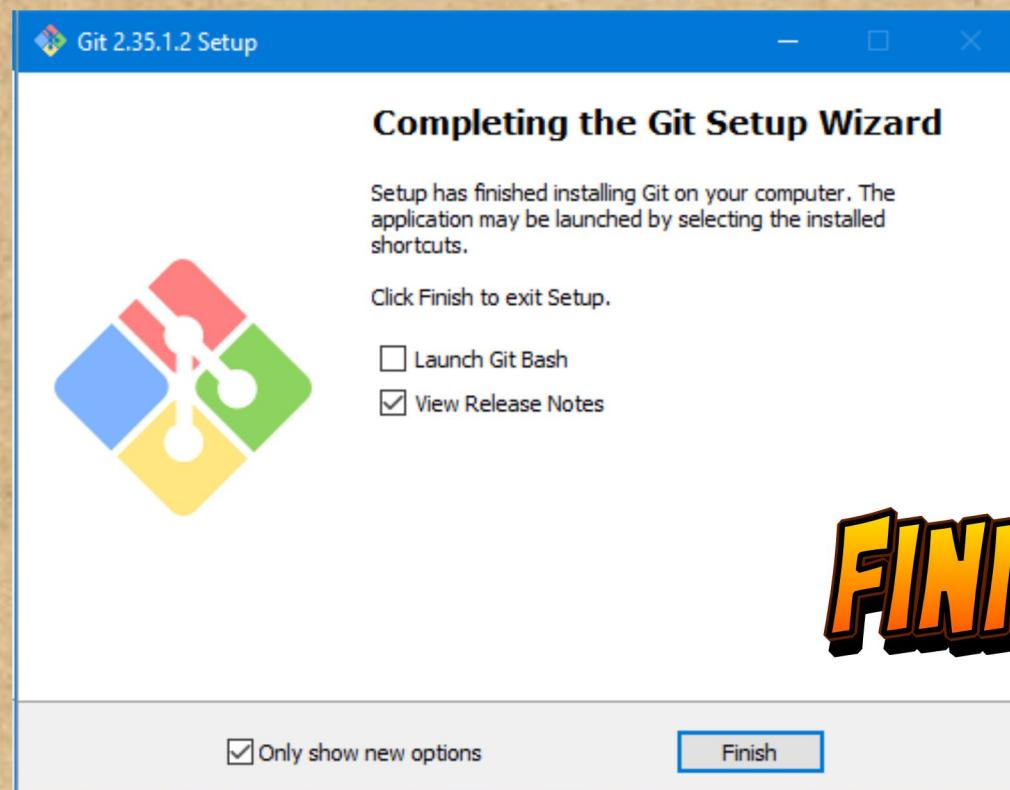


Si utilizas el navegador google chrome puedes ejecutar directamente el instalador, sino puedes ir a la carpeta de descargas y le das doble clic al instalador (.exe)





Si todo nos ha ido super bien, ya tendremos instalado git



← → C Archivo | C:/Program%20Files/Git/ReleaseNotes.html

HOME PAGE FAQ GIT PRIVATE BUGS QUESTIONS

## Git for Windows v2.35.1(2) Release Notes

Latest update: February 1st 2022

### Introduction

These release notes describe issues specific to the Git for Windows release. The release notes covering the history of the core git commands can be found [in the Git project](#). See <http://git-scm.com/> for further details about Git including ports to other operating systems. Git for Windows is hosted at <https://gitforwindows.org/>.

#### Known issues

Should you encounter other problems, please first search [the bug tracker](#) (also look at the closed issues) and [the mailing list](#), chances are that the problem was reported already. Also make sure that you use an up to date Git for Windows version (or a [current snapshot build](#)). If it has not been reported, please follow [our bug reporting guidelines](#) and report the bug.

#### Licenses

#### Changes in v2.35.1(2) since v2.35.1 (January 29th 2022)

#### Bug Fixes

- A bug in FSCache that triggered by a patch that made it into Git for Windows v2.35.0 was fixed.

#### Changes in v2.35.1 since v2.35.0 (January 24th 2022)

#### Changes in v2.35.0 since v2.34.1 (November 25th 2021)

#### Changes in v2.34.1



Luego de dar clic en la opción finish por lo general se nos abre el navegador y nos muestra las novedades de Git, pero esto solo es a manera informativa.



Muy bien el proceso de instalación en windows es muy facil, solo debes dar siguiente, siguiente, siguiente ... hasta llegar al final

Para Comprobar versión de Git solo basta ir al menú inicio, escribes CMD y luego le das clic (puede llamarse simbolo del sistema), posteriormente se abrirá una pantalla negra y digitas git --version y das enter

```
Microsoft Windows [Versión 10.0.18363.2094]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\SENA>
```

```
Microsoft Windows [Versión 10.0.18363.2094]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\SENA>git --version
git version 2.29.2.windows.2

C:\Users\SENA>
```



En este momento ya tienes git instalado, muy bien haz ganado una rosa por tu esfuerzo

# GITHUB



Para adquirir los servicios de GitHub, es necesario acceder al sitio web:  
[www.github.co](https://www.github.com)

El software Github, es un software que permite el almacenamiento de proyectos para trabajo colaborativo a través de un sistema de control de versiones a través de Git.

Esta herramienta es ampliamente empleada en el campo del desarrollo de software, permitiendo un constante proceso de actualización, mejora y adaptación de los cambios hechos al proyecto trabajado.

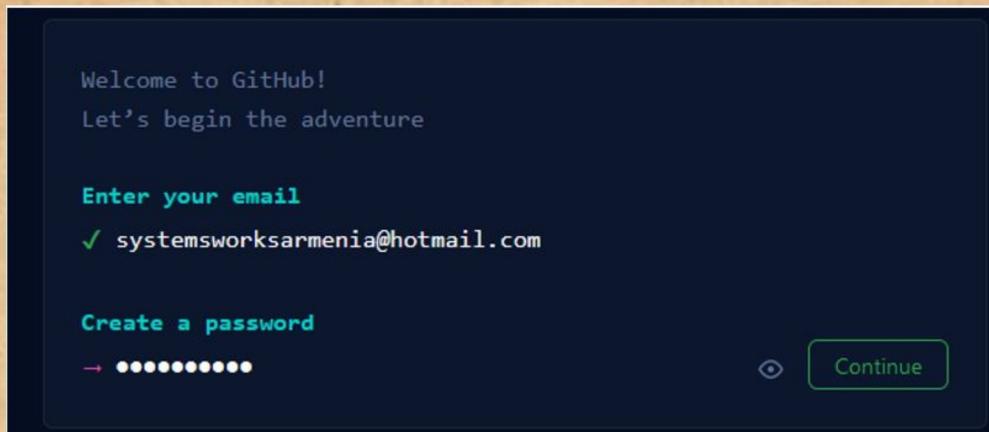
Permite con el manejo de versiones, correcciones del software, salvar prototipos en versiones funcionales y beta para su posterior desarrollo y adaptabilidad a las necesidades del negocio.

Debes dar clic en Sign Up

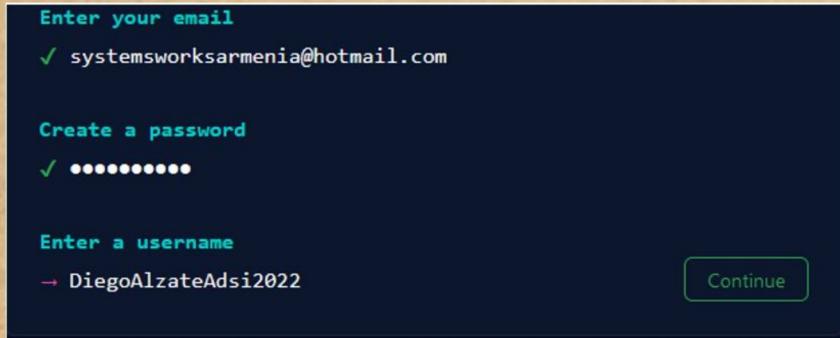
Al momento de ingresar, el sistema te pregunta si tienes un usuario registrado. Esta comprobación la realiza mediante el correo electrónico



Una vez hecha la comprobación y permita el acceso, debes crear tu contraseña. Esta debe contener letras Mayúsculas, minúsculas y números para efectos de aumentar la seguridad.



Después de la creación de la contraseña, el sistema indica que debe crear un nombre de usuario para que sea identificado dentro de la comunidad GitHub, así como su entorno de trabajo



Y si todo ha salido bien ... Te haz ganado una rosa, por tu esfuerzo hasta aqui ... Luego de hacer todo el proceso, ya tenemos cuenta en GitHub

# ESPAZO DE TRABAJO

The screenshot shows a GitHub profile page titled "Perfil de usuario". At the top, there's a search bar and navigation links for "Pull requests", "Issues", "Marketplace", and "Explore". Below the header, there's a sidebar with "Recent Repositories" and "Recent activity". A large green button labeled "New" is prominently displayed. Two arrows point from the text below to the "New" button and the user profile area respectively.

Recent Repositories

Find a repository... New

Diego780527/Deploy  
Diego780527/copia\_deploy  
Diego780527/ADSI

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide Start a project

Join GitHub Global Campus!

Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community.

Perfil de usuario

Listado de repositorios recientes del usuario



Manos a la obra, crearemos un repositorio: Una vez dentro de la plataforma GitHub, buscamos el botón nuevo, para iniciar la creación de un nuevo repositorio

Luego de dar clic en nuevo, aparece una ventana en donde se piden los datos del repositorio, nombre, el propietario que para el caso es la cuenta creada, la descripción del repo y el nivel de seguridad, si es de carácter público o privado

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner \*

Repository name \*

Diego780527 /

Great repository names are short and memorable. Need inspiration? How about [super-octo-engine](#)?

Description (optional)

Public

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

Private



Ahora bien... Al terminar de procesar los datos queda de la siguiente forma:

Owner \* Repository name \*

Diego780527 / TesterAdsi ✓

Great repository names are short and memorable. Need inspiration? How about [super-octo-engine?](#)

Description (optional)  
glosa

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:  
Skip this step if you're importing an existing repository.

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

Add .gitignore  
Choose which files not to track from a list of templates. [Learn more](#).

Choose a license  
A license tells others what they can and can't do with your code. [Learn more](#).

Luego dar clic en crear repositorio Create repository

main 1 branch 0 tags

Go to file Add file Code

Diego780527 Update Profle ... 9e3fa44 5 days ago 3 commits

static	Despligue Angular ok	21 days ago
templates	Despligue Angular ok	21 days ago
Profile	Update Profle	15 days ago
main.py	Despligue Angular ok	21 days ago
requirements.txt	Despligue Angular ok	21 days ago

Help people interested in this repository understand your project by adding a README. Add a README



Ahora procedemos a anexar documentos a versionar... A través del botón Add file, se procede a anexar los archivos

# CONEXIÓN ENTRE GIT Y GITHUB

Antes de hacer conexiones, debemos verificar que el usuario tenga cuenta en Git, para la realización del proceso, de ser exitoso procedemos a los siguientes pasos:

1. Autenticación de Git en GitHub
  - Conectar por HTTPS

Es el procedimiento más recomendado, ya que permite almacenar las credenciales en el caché de Git por medio de un asistente de credenciales.

- Conectar por SSH

Al ejecutar el proceso de autenticación (clonación) por SSH, se debe generar claves SSH en cada equipo a trabajar desde GitHub Enterprise Server.

2. Conexión Exitosa

Nota: si presenta algún problema al respecto, revisar si las credenciales son correctas o tiene permisos del administrador de red para el manejo de la herramienta.



# ELABORADO POR LOS MEJORES

DIEGO ALZATE

ALEJANDRO GIRALDO

JACOB HOYOS

DIANA RESTREPO

JULIAN FRANCO



# GIT Y GITHUB

---



HOLA, EN ESTE MODULO  
VEREMOS LA CONFIGURACION SSH  
EN GIT Y GITHUB Y COMO INVITAR  
USUARIOS EN GITHUB



# MODULO 3

# SH E INVITACIÓN

# **OBTENER Y CONFIGURAR TUS CLAVES SSH PARA GITHUB**



EN ESTE MOMENTO  
ESTAMOS INGRESANDO  
A UN NIVEL INTERMEDIO  
ALTO. DONDE SE  
APLICARAN CONCEPTOS  
DE TRABAJOS  
COLABORATIVOS

# SSH



Que es SSH?

El protocolo SSH es un sistema de comunicación segura. Significa que en este tipo de transmisión la información va encriptada. Como ya hemos dicho, todos los protocolos se dividen en capas dependiendo de la función y finalidad de estos. En este caso, el protocolo SSH actúa en la capa de aplicación. Con todo esto, podemos deducir que el protocolo SSH es un sistema de acceso remoto al host que tenemos contratado de manera segura.



Pero, ¿qué es SSH? SSH se traduce como Secure Shell o, dicho de otro modo, terminal seguro. Permite que el usuario pueda acceder, modificar y administrar su servidor de forma totalmente segura y de manera remota

Primero, comprueba si ya has generado las claves SSH para tu máquina. Abre una terminal e introduce el siguiente comando, ssh -v Muestra si se tiene una llave creada en el en el equipo



ssh -v

```
C:\Windows\system32\cmd.exe
C:\Users\User>ssh -v
usage: ssh [-46AaCfGgKkMNqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
C:\Users\User>
```

# CREACIÓN DE LLAVE SSH

Si no se tiene una llave ssh en el equipo, la debemos crear

```
□ Seleccionar C:\Windows\system32\cmd.exe  
D:\1. Git>
```



Primero Seleccionamos el directorio donde almacenaremos las llaves públicas y privada

Estando aquí desplegaremos el comando para crear la llave

ssh-keygen



```
D:\1. Git>CD 0.SSH
D:\1. Git\0.SSH>
D:\1. Git\0.SSH>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\User/.ssh/id_rsa): clave_ssh
Enter passphrase (empty for no passphrase).
Enter same passphrase again:
Your identification has been saved in clave_ssh.
Your public key has been saved in clave_ssh.pub.
The key fingerprint is:
SHA256:XRQrwsNokIIXQdCrEwiftMZ+X/70yg1DHD4VvBwvJjA user@DESKTOP-K80V34C
The key's randomart image is:
+---[RSA 3072]---+
|.*+o.. .+.
|o B .. +E .oo
|oB + o =oo.++
|o * .. *.== .
|o . . oS *o .
|o . . . .
|. oo.
|..+.
|o..
+---[SHA256]---+
D:\1. Git\0.SSH>
```

Ahora bien, aquí tenemos...

- A. El Comando ssh para crear llave
- B. Es donde nombramos el archivo(si no lo nombramos quedara por defecto id\_rsa)
- C. En este campo se digita una contraseña para cifrar y generar el archivo donde se contendrá la llave.
- D. Introducción de la contraseña para la contraseña para ser verificada.
- E. Clave ssh generada



```
C:\ Seleccionar C:\Windows\system32\cmd.exe
| ..+.
|   o..
+---[SHA256]----+
D:\1. Git\0.SSH>dir
El volumen de la unidad D no tiene etiqueta.
El n mero de serie del volumen es: FED7-0824

Directorio de D:\1. Git\0.SSH

04/03/2022 10:47 a. m.    <DIR> .
04/03/2022 10:47 a. m.    <DIR> ..
04/03/2022 10:47 a. m.          2.655 clave_ssh
04/03/2022 10:47 a. m.          575 clave_ssh.pub
                2 archivos           3.230 bytes
                2 dirs  93.820.198.912 bytes libres

D:\1. Git\0.SSH>
```

A. Verificamos que se crearan 2 archivos uno es la clave p blica y el otro es la clave privada.



El contenido de los archivos que tenemos  
es ...

## Clave pública

```
El modo restringido está destinado a la navegación segura por el código. Confíe en esta ventana para habilitar todas las funciones. Administrar Más información

D: > 1. Git > 0.SSH > clave_ssh.pub

1 ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDeY
+GRQPiAJ8sgOwIQQopoXreAxvGDSNe8G52qATAEgxDYEsjOhXukXI54TzDj7CIxmo4jZzG0FNLoLHSC2eazphOP6B6fCHS9skV7ZFKCwzVukf55ewbCqUqJB5IjHEfp/wA5vzYXcA4ct8YfRduB/
sjveTyLYR3xA3N1B34iYw/uwVhrFFuLax04g6UcQz0UEBywzhi37r0khVtUawtuiKXar6WF8F7D0Dh1nC/aepvhrlAbdufkvD9jIXQpxAp+QP3y95si861P51sm9MAFLu/
NJRAJAdz31tHFekedckLDYvinqEMWlxMqtUpIso0XCTRIBAvlsunEb7Yajqlxh61r0aTSN2T6s5dNL4qp5m2dxv+OsnmBzsnnck/
8x132bMcyWjHogcvxy8p2pVpS6FDamvE50HmpbzEe8pjGA4Ti8lq/+3/SZxDc0/q4pLf6fLjxtc/6xDG2zwov2R3PfL1vrC1A2xAec7drjc26YpgW6L3hzg+L19RNU= user@DESKTOP-K80V34c
```

## Clave privada

```
D: > 1. Git > 0.SSH > clave_ssh

1 - - - BEGIN OPENSSH PRIVATE KEY - - -
2 b3B1bnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABDc6f3/r3
3 iUiCPnR5XoMZXA AAAEAAAAAAGXAAAAB3NzaC1yc2EAAAQABAAQDeY+GRQPiA
4 J8sgOwIQQopoXreAxvGDSNe8G52qATAEgxDYEsjOhXukXI54TzDj7CIxmo4jZzG0FNLoL7
5 HSC2eazphOP6B6fCHS9skV7ZFKCwzVukf55ewbCqUqJB5IjHEfp/wA5vzYXcA4ct8YfRdu
6 B/sjveTyLYR3xA3N1B34iYw/uwVhrFFuLax04g6UcQz0UEBywzhi37r0khVtUawtuiKXar
7 6WF8F7D0Dh1nC/aepvhrlAbdufkvD9jIXQpxAp+QP3y95si861P51sm9MAFLu/NJRAJAd
8 z31tHFekedckLDYvinqEMWlxMqtUpIso0XCTRIBAvlsunEb7Yajqlxh61r0aTSN2T6s5d
9 NL4qp5m2dxv+OsnmBzsnnck/8x132bMcyWjHogcvxy8p2pVpS6FDamvE50HmpbzEe8p
10 jGA4Ti8lq/+3/SZxDc0/q4pLf6fLjxtc/6xDG2zwov2R3PfL1vrC1A2xAec7drjc26YpgW
11 6L3hzg+L19RNUAAwQKbxKs1hxVc64dyg1+pAuMo4n8FaLGrs5+Mlgxn8tjRZGRqq5rovEC
12 /ff4uruKZwTdpSoif0sr2vdRBnHcEjyiuoDXa1k0R02JeJAe0K1YcBY4pT2ZsmFe8h+ABY
13 E7NjG/xuppSTThrsWd3ffz7bJOMKKTzd5ne+cHTffeKhs72t7ukU59Aq93F3XsVadvEqCY
14 CtN03yKXw42aCnTjls61GoZfZELsV0y507G+63aQqLGQt/QZTdW91N4vpCTfEX3mu9hmv/
15 pF85j0QPU1QiEq6sIyB8DpfB2KpjdB3U3op8D0718ztMP51gnik3s0EPckghuB53v95Dp1
16 VhUC2qIzejUh/XOp3/WDUkYrJwoEEMJLFYJGPrfut3CLYtVr1tQ4DWCL2SR/OiXyNL2U9/
17 K6zIPyQV5gKiGd11AhcxipDQ9oB/oytmKt2ZXXxnXgR2J2Rcg1pWSSzJ0vPRBBSSQMj7dE1
18 SabxzD6qC0vlboNSU70imqW4XvjVd7Wlk6kj15d+DjIi2/1A11dqsf8JFyDvVzgRQ14n2
19 3iQUn6Y+YNNex480xPC6H1OGApmZxuwqoHNTfhKVxt7KmVxt0v/lvP0+53exJDjfHWRk6r
20 MmuJz+w39pMNNkR0M9b9R3jX7IsMkFSE9vWCCOKf4EPoedUXUMYnKIEZPwVuPiD42566vm
21 Lf1CkjMbDeGHE5P0T+Wa8IFPpuYtDjTghc3aeulBUDkp0s/WfbPMk3/yrVWeiIYHb17Rg
22 81DS6/zexM1GRq28WYzWgPif0fbfgAY1Ve7KGNgNdOUjIOyV96IxuUPg58P8qC9hDr3dDj
23 p0eaFG5P1vAFU02YoCAFvAevdzMFwkvK0geBL7Z4aG4tI6s371I2Izb1r7lcpqrfb451Y
24 a6NyGL50k91ENbp0kyWQpuvghwt6+TREssVLKEJIueFrhr9z0k7Aip3j5LWZTEUF5yGDmx
25 Z/dvoT/Bc2e3j00LXX4kUMATNKVoVMjaYFZhUjtTkC+5zXhMCJRPD1EB1iSJ4C51QcknfWg
26 FwqhMIrcG3z+jexfY4QxioE4pyRn2FpUnX5aADQrI5Zr9z9N9L1XATL+qTc3gMZCto7Q6E
27 SpxNq/WLP1N1MgnJeBZgMHerpzY1y1BtrcYQ3q1DKeIetkIN4Cut6MyisxeVRI0qKKp2g6
28 e9RJ1Zjr92P1kJMxiF7IdTNeqW5eMx/C1LLD9UCBuBzHs85kY1IZUXxMNE00387F+0YIfy
29 mOsZabIe0rkUXVo8W3TjWD8wQrdWciW3mU4U3spffiOfM8xsQ0TwHgF0trQW8Nt03P7I1+
30 iHdoptciQqvqcD00+lw092jjf127meBv3PKcKOsyBB1mlkTVzu0x1/CFpDJ5JvS4WA1nm
31 LcNEtMBigwBiXRieUG0LCUiPTgxPrkXpwkZYbqFED01og0ERHAgfDY/+EZtvGRjj0i5LYt
32 XyrYY+15qnll+ghTyaPLe/h6monq3cZ2BEIIIMpQvai12I27pb5cwLiSAgbPwjYQbeyXxA
33 NjJLzB/M4MQTIuFIIdI0usmsjKGTJ8dWzsqqDFTHiyr7uDY1KG3vg7uKNAZUeDvJ90WCvba
34 YUq+mcRSCTALhMb/ur86wyx39GnM002jH61A7DyD2bR3WuD235q1XVfkBqzLBThbmsyF+b
35 MsJ+7x11167WUoRmAvAYJP1eH0w3sOpoGH6i8q5/LbtBci4iViekZSyKZmohjXp1W3t+uB
36 VU07t1ewrh7FpVuTPNYaE6WvxIApoTjchcqwxka3AAqMop2QmennacMhg9yWhm2whd8r5x
37 /G7/jMrj+qtOceRd8L7mNRZ1nFXB/xYY7813dP13K7/2wkv5Ts0dtkIHawpntdjt1/f9gR
38 B6HCvgL5AQwLudE5S/EwjQBx+c4=
39 - - - END OPENSSH PRIVATE KEY - - -
```

Despues de tener las claves creadas nos dirigimos a nuestro repositorio en github para crear la conexión a nuestro equipo.

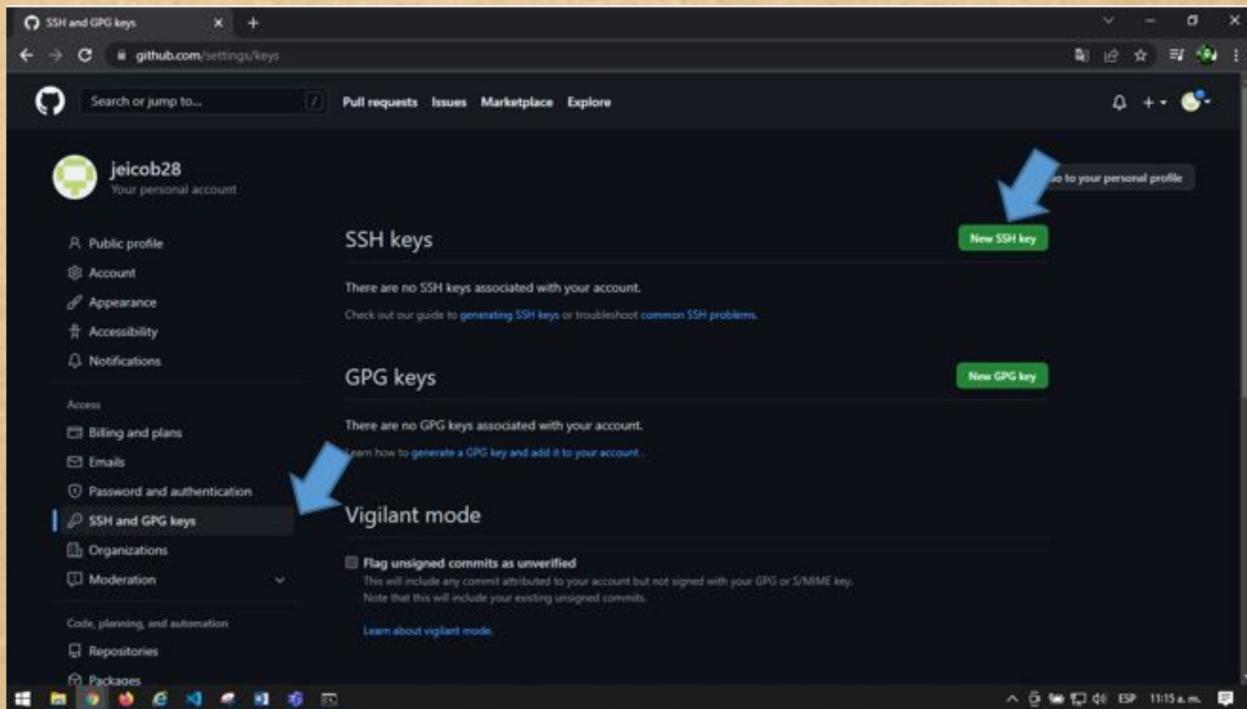


Despues de tener las claves creadas nos dirigimos a nuestro repositorio en github para crear la conexión a nuestro equipo.

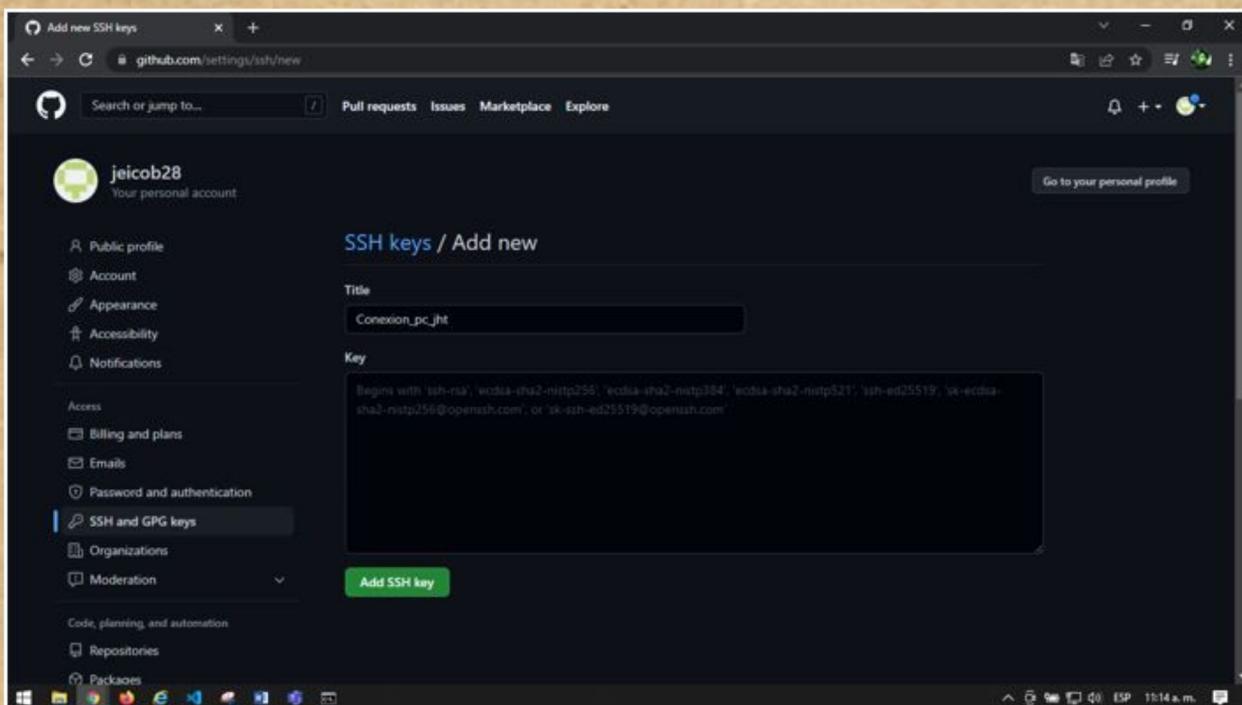
## 1. Nos dirigimos a la configuración de a cuenta

The screenshot shows a GitHub profile for the user 'jeicob28'. The sidebar on the right includes links for 'Your profile', 'Your repositories', 'Your codespaces', 'Your projects', 'Your stars', 'Your gists', 'Upgrade', 'Feature preview', 'Help', and 'Settings'. The 'Settings' link is highlighted with a blue arrow. The main content area displays the user's activity feed, recent repositories ('jeicob28/jeicob28', 'jeicob28/repo'), and a 'Discover interesting projects and people' section.

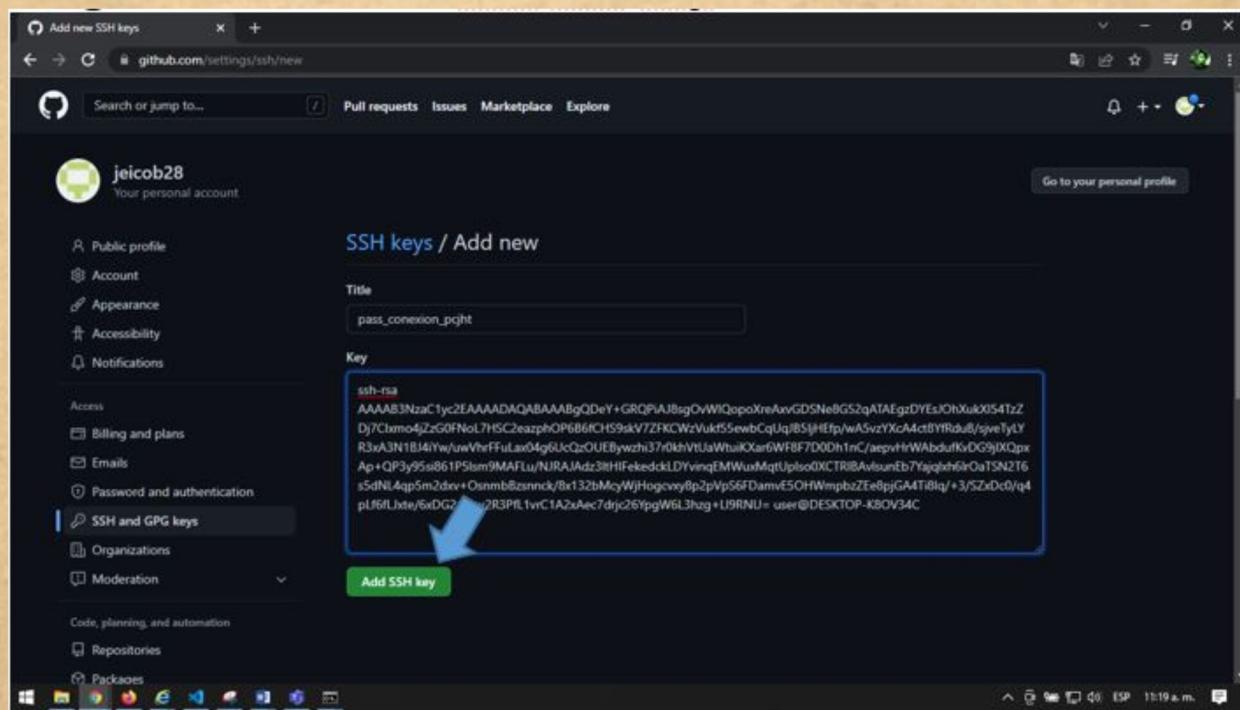
2. Luego pulsamos en ssh and GPGKeys y damos clic en New ssh key



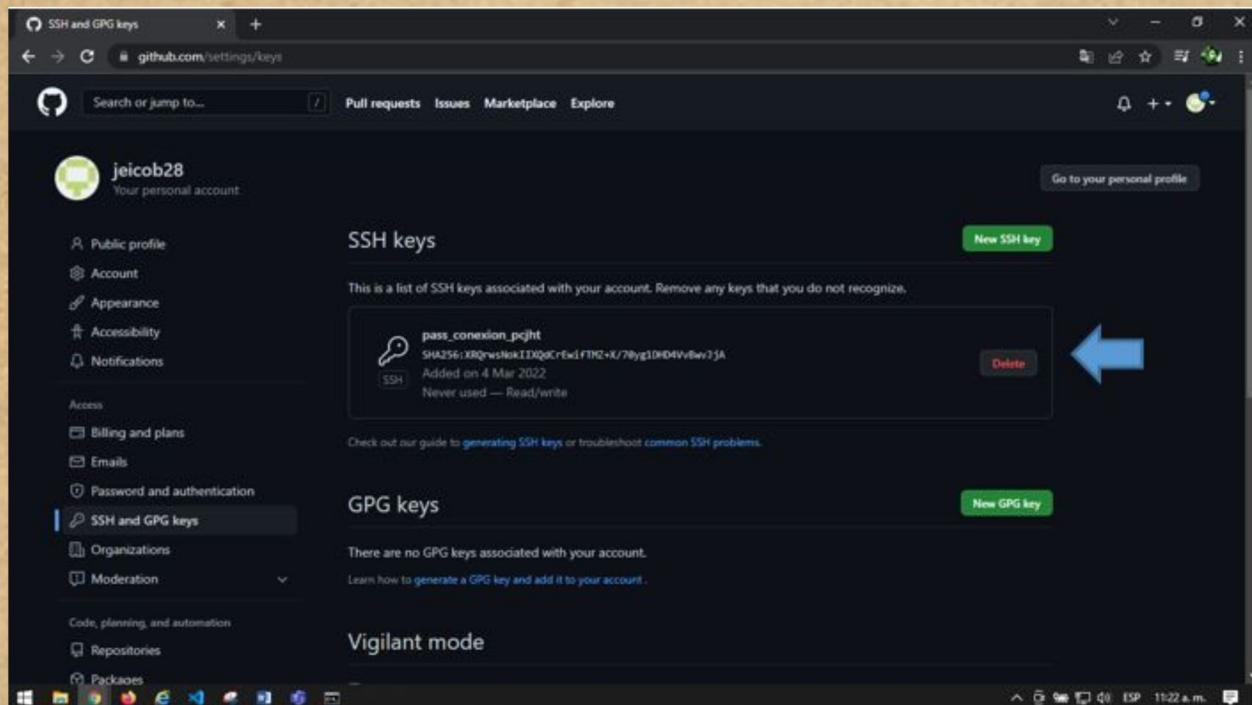
3. Luego damos en ssh key, colocamos un nombre de referencia y la clave publica



#### 4. Luego damos clic en add ssh key



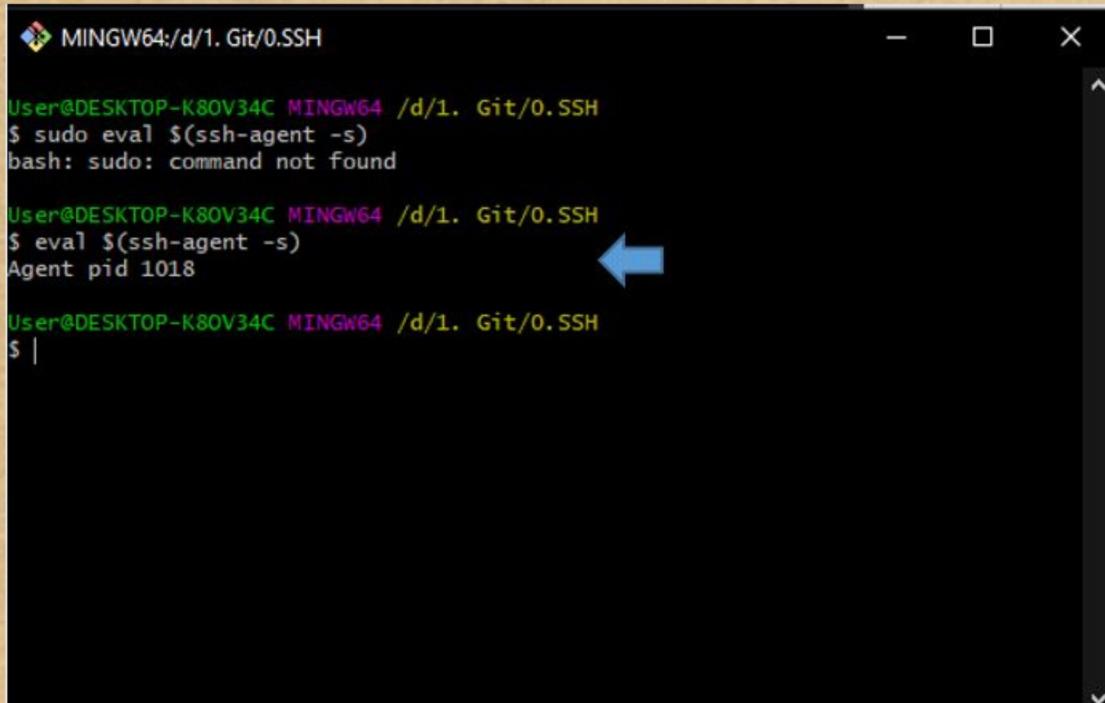
Después de agregar la clave que cargada



Posterior a este proceso ya tenemos que agregar la clave ssh a nuestro equipo, para ello ejecutamos una ventana de comandos, se recuerda que al inicio cuando se instala github en Windows también se instala la ventana de comando de bash de Linux que nos permite ejecutar comandos de este sistema operativo.



Inicializamos una ventana de comando de bash,  
e iniciamos el cliente ssh



```
MINGW64:/d/1. Git/0.SSH
User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ sudo eval $(ssh-agent -s)
bash: sudo: command not found

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ eval $(ssh-agent -s)
Agent pid 1018
←
User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ |
```

Con el comando ssh-add ya que ejecutamos la ventana de bash en la misma ruta donde se encuentra el archivo con la clave privada.



```
MINGW64:/d/1. Git/0.SSH
User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ sudo eval $(ssh-agent -s)
bash: sudo: command not found

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ eval $(ssh-agent -s)
Agent pid 1018

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ssh-add D:\1. Git\0.SSH\clave_ssh
D:\1.: No such file or directory
Git0.SSHclave_ssh: No such file or directory

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ pwd
/d/1. Git/0.SSH

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ls
clave_ssh  clave_ssh.pub

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ssh-add clave_ssh|
```

Nos pedirá la clave con la que ciframos el archivo



```
MINGW64:/d/1. Git/0.SSH
User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ eval $(ssh-agent -s)
Agent pid 1018

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ssh-add D:\1. Git\0.SSH\clave_ssh
D:\1.: No such file or directory
Git0.SSHclave_ssh: No such file or directory

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ pwd
/d/1. Git/0.SSH

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ls
clave_ssh  clave_ssh.pub

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ssh-add clave_ssh
clave_ssh      clave_ssh.pub

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ssh-add clave_ssh
Enter passphrase for clave_ssh: |
```



Si todo nos sale bien, Nos dará la identificación del equipo y quedara agregada la clave, aqui mereces una rosa como reconocimiento a tu esfuerzo y dedicación

```
◆ MINGW64:/d/1. Git/0.SSH
User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ssh-add D:\1. Git\0.SSH\clave_ssh
D:\1.: No such file or directory
Git0.SSHclave_ssh: No such file or directory

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ pwd
/d/1. Git/0.SSH

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ls
clave_ssh  clave_ssh.pub

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ssh-add clave_ssh
clave_ssh      clave_ssh.pub

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ ssh-add clave_ssh
Enter passphrase for clave_ssh:
Identity added: clave_ssh (user@DESKTOP-K80V34C)

User@DESKTOP-K80V34C MINGW64 /d/1. Git/0.SSH
$ |
```

# **INVITAR A 3 PERSONAS AL REPOSITORIO**

Es un proceso sencillo  
Se inicia sesión en el repositorio  
en línea de githut.



Seleccionamos el repositorio



Search or jump to... Pull requests Issues Marketplace Explore

Recent Repositories [New](#)

Find a repository... [jeicob28/jeicob28](#) [jeicob28/repo](#)

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

All activity

Introduce yourself

The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

jeicob28 / README.md

1 - 🌟 Hi, I'm @jeicob28  
2 - 💬 I'm interested in ...  
3 - 🚧 I'm currently learning ...  
4 - 🤝 I'm looking to collaborate on ...  
5 - 📩 How to reach me ...

Dismiss this Continue

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch or star and people you follow.

Posteriormente a settings o configuracion

Search or jump to... Pull requests Issues Marketplace Explore

jeicob28 / jeicob28 Public

Code Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

jeicob28 este es el primer comentario 4 days ago 1 commit

PrimerText.txt este es el primer comentario 4 days ago

jeicob28/jeicob28 is a special repository. Its README.md will appear on your public profile!

Add a README

## Luego en collaborators

A screenshot of a GitHub repository settings page. The repository name is 'jeicob28 / jeicob28'. The 'Settings' tab is active. On the left, there's a sidebar with 'General' selected. Under 'Access', the 'Collaborators' option is highlighted with a blue arrow pointing to it. Other options like 'Moderation options', 'Branches', 'Tags', 'Actions', and 'Webhooks' are also listed. The main area shows the repository name 'jeicob28' and a 'Social preview' section.

Se visualizara la configuración del repositorio, para agregar o invitar un colaborador damos clic en addpoeple



A screenshot of the 'Who has access' section in GitHub settings. The 'Collaborators' tab is selected in the sidebar. The main area shows that the repository is a 'PUBLIC REPOSITORY' and that there are '0 collaborator' with 'DIRECT ACCESS'. A 'Manage' button is available. Below this, the 'Manage access' section shows a message: 'You haven't invited any collaborators yet' and a green 'Add people' button, which is highlighted with a blue arrow.

Se desplegará  
una ventana  
donde  
colocaremos el  
email de las  
personas



Projects Security Insights Settings

Who can view this repository

PUBLIC

This repository can be viewed by anyone with the link.

Manage access

Add a collaborator to jeicob28

Search by username, full name, or email

Select a collaborator above

A blue arrow points to the search bar.



Si todo sale bien, Asi que darían cargadas las personas invitadas, aqui mereces un reconocimiento por tu esfuerzo

### Manage access

Add people

Select all Type ▾

Find a collaborator...

<input type="checkbox"/>	 @carorestrepo	Awaiting 45ca's response	Pending Invite <input type="button" value="Accept"/>	<input type="button" value="Delete"/>
<input type="checkbox"/>	 DiegoM	Awaiting Diego780527's response	Pending Invite <input type="button" value="Accept"/>	<input type="button" value="Delete"/>
<input type="checkbox"/>	 jdfrancop	Awaiting jdfrancop's response	Pending Invite <input type="button" value="Accept"/>	<input type="button" value="Delete"/>

Get team access controls and discussions for your contributors in an organization.

**NEW** Private repos and unlimited members are free.

Create an organization

# ELABORADO POR LOS MEJORES

DIEGO ALZATE

ALEJANDRO GIRALDO

JACOB HOYOS

DIANA RESTREPO

JULIAN FRANCO



# GIT Y GITHUB

---



HOLA, EN ESTE MODULO  
APRENDEREMOS A CREAR UN  
COMMIT Y COMO REGRESAR A  
ESTE DE SER NECESARIO



# MODULO 4

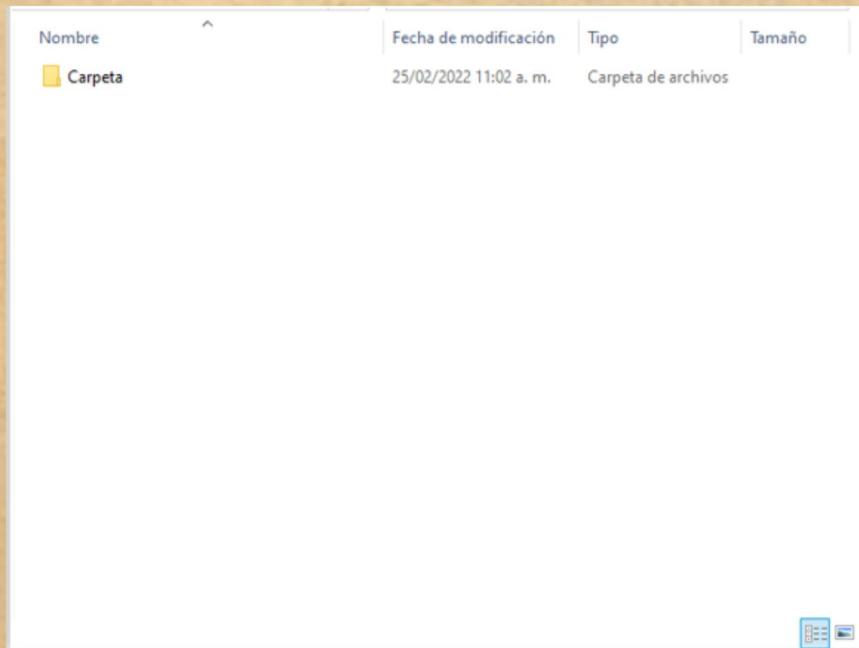
# COMMITs



# MANOS

# A LA OBRA

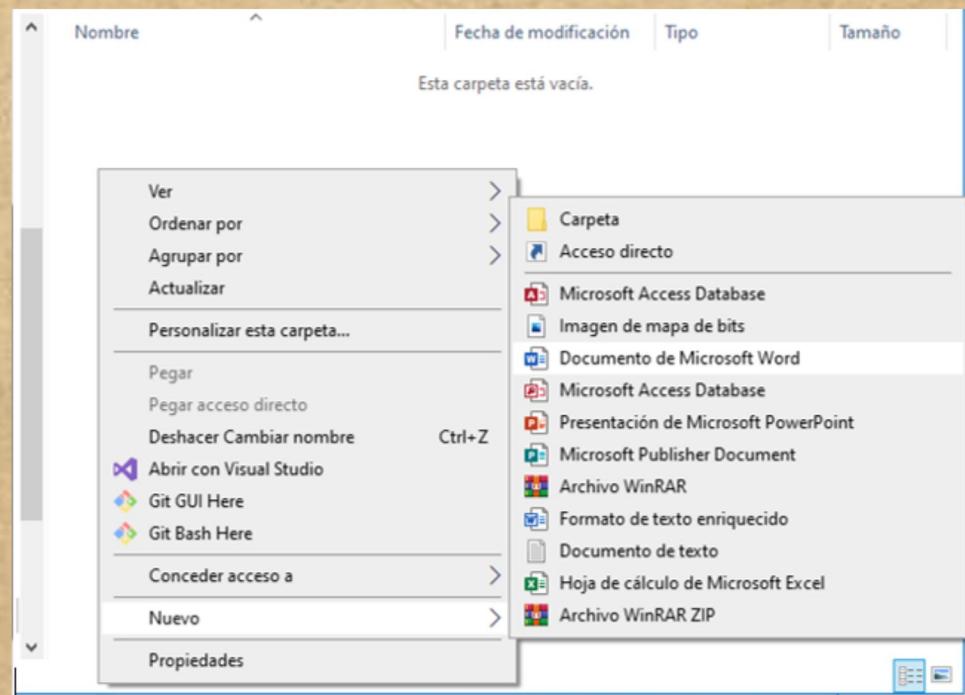
# CREAR CARPETA - DOCUMENTO

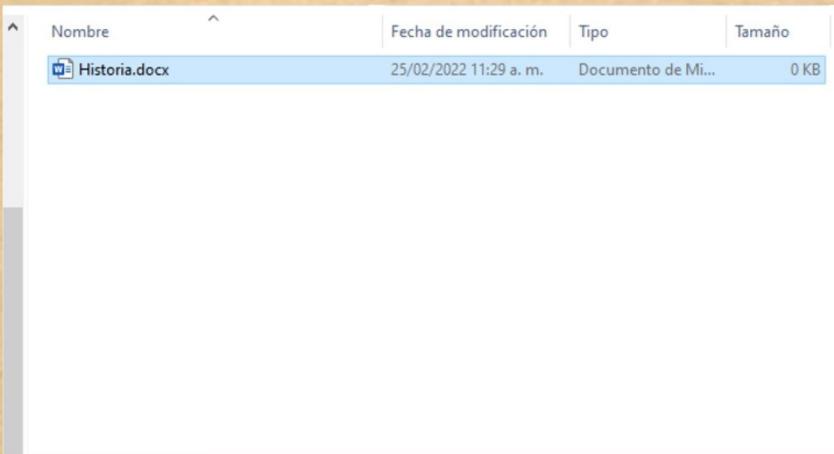


Lo primero que vamos a hacer es crear una carpeta en cualquier ubicación de tu computadora



Luego crearemos un archivo de Word, dentro de esta carpeta, puedes hacerlo usando el clic derecho del mouse dentro del espacio blanco de la carpeta que has creado, luego te posicionas en nuevo y posteriormente en documento de Microsoft Word





Luego le asignas un nombre que te guste, no limites tu creatividad



A screenshot of a Microsoft Word document titled 'Historia.docx'. The ribbon menu is visible at the top. The main content area contains the text: 'Muy buenos días, esta es la historia de como aprenderemos Git y GitHub'.

Ahora realizaremos el primer cambio,  
en este caso vas a escribir una linea  
en tu documento de Word



Espero que hasta aquí no hayas tenido ningún inconveniente, te has ganado la primera rosa de reconocimiento por tu esfuerzo



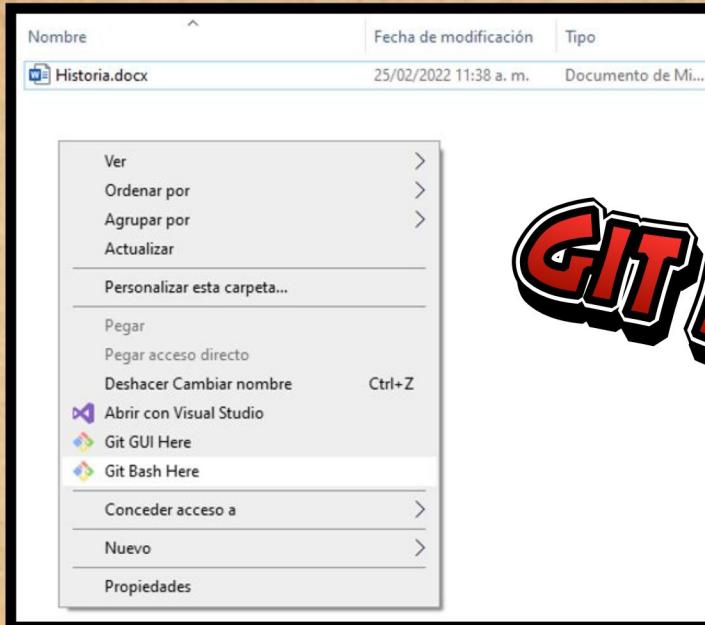
En este momento ya empezaremos a trabajar directamente en Git, ya debes tener instalada la herramienta y configurado tu nombre de usuario y correo electrónico, si no es así, regresa un poco en este manual.

```
E:\EJEMPLO\Carpetas>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.http://dav.ejemplo.com.usertopath=true
user.name=aquitunombre
user.email=tucorreoelectrónico@ejemplo
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
credential.helper=store
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
E:\EJEMPLO\Carpetas>
```

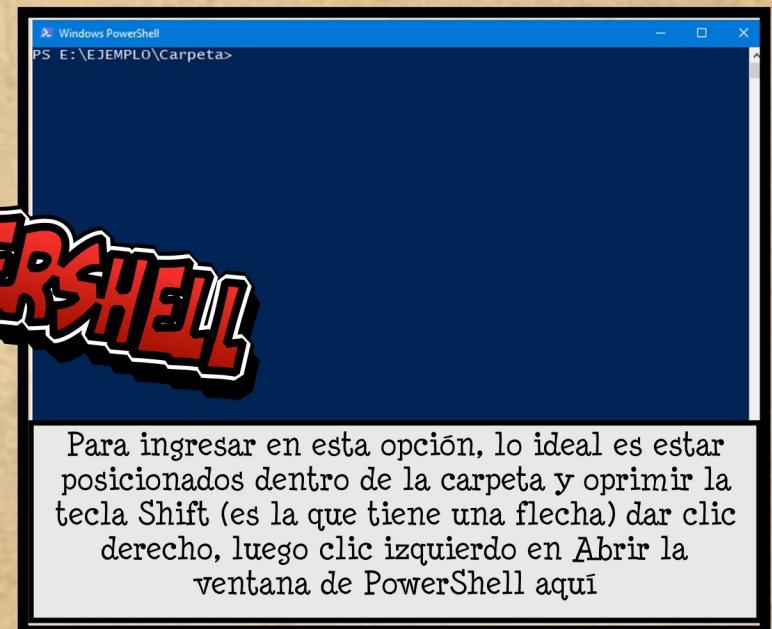
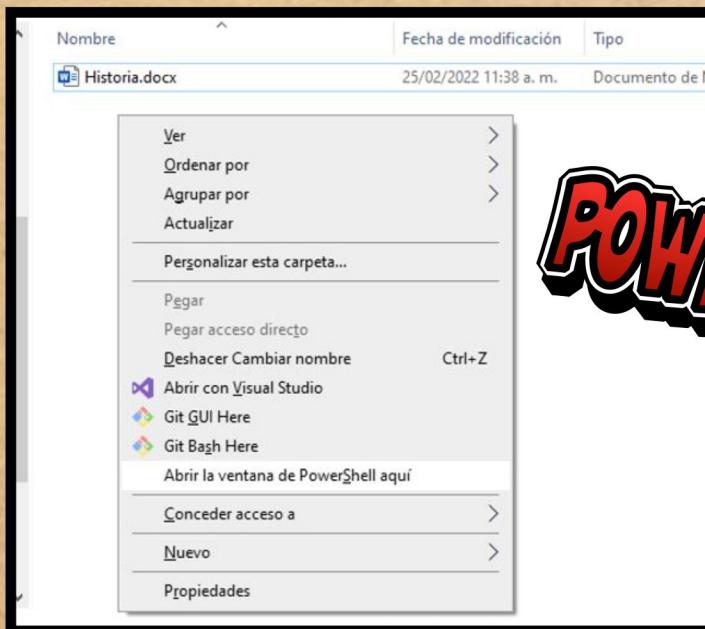
A screenshot of a Windows command prompt window titled "Símbolo del sistema". The window shows the output of the command "git config --list". A red rectangular box highlights the "user.name" and "user.email" configuration lines. A red arrow points from the bottom of this box to the "user.name" line, indicating it as the target for configuration.



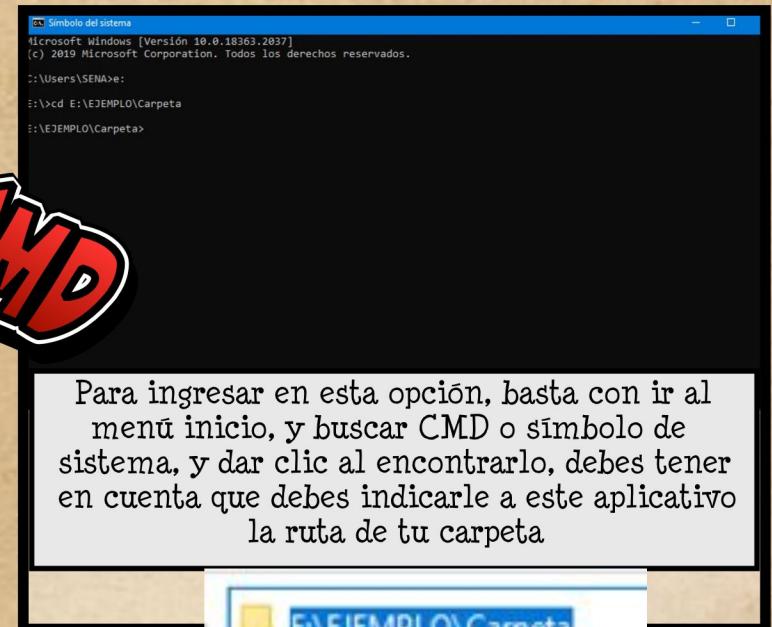
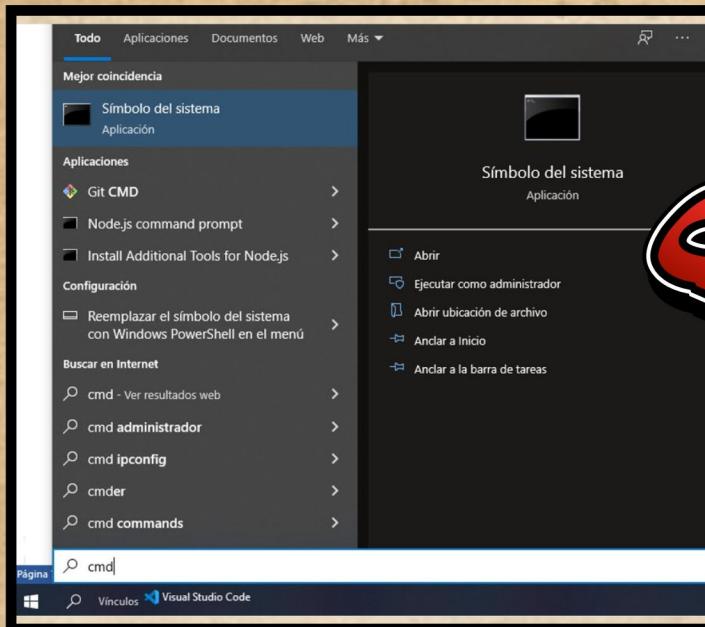
Vas a poder trabajar en diferentes aplicativos... para este momento puedes utilizar el CMD (símbolo de sistema), el Powershell de Windows, el Git Bash (herramienta que se instala al instalar el Git)



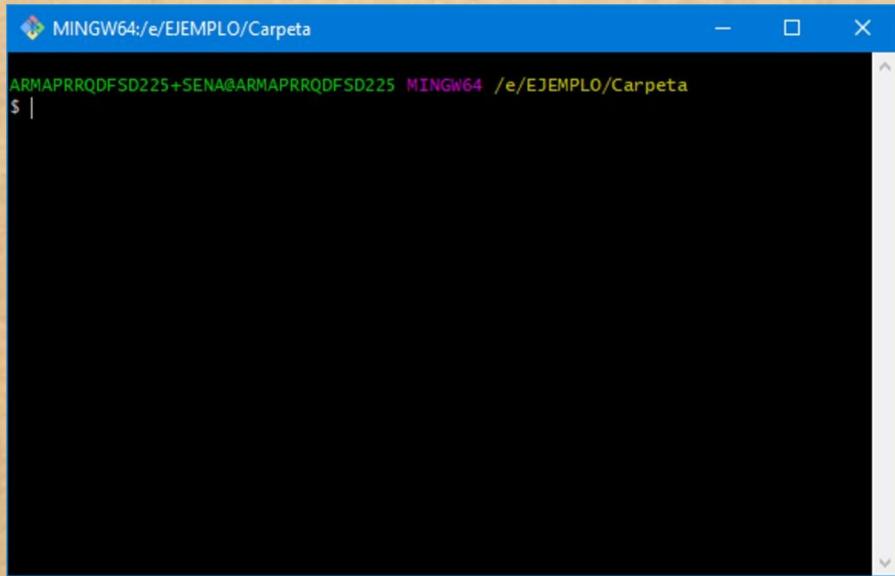
Para ingresar en esta opción, lo ideal es estar posicionados dentro de la carpeta y dar clic derecho, luego clic izquierdo en Git Bash Here



Para ingresar en esta opción, lo ideal es estar posicionados dentro de la carpeta y oprimir la tecla Shift (es la que tiene una flecha) dar clic derecho, luego clic izquierdo en Abrir la ventana de PowerShell aquí



Para ingresar en esta opción, basta con ir al menú inicio, y buscar CMD o símbolo de sistema, y dar clic al encontrarlo, debes tener en cuenta que debes indicarle a este aplicativo la ruta de tu carpeta



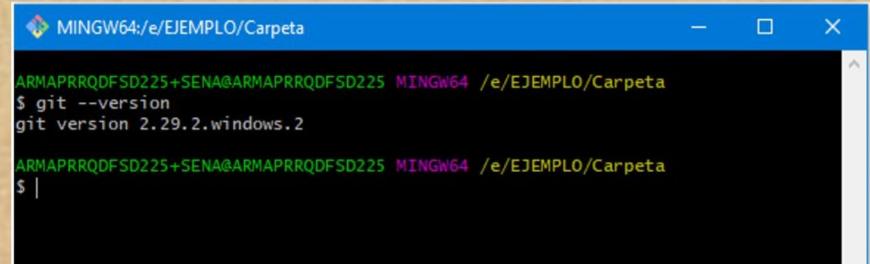
```
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta
$ |
```



Yo he decidido utilizar el Git Bash



git --version



```
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta
$ git --version
git version 2.29.2.windows.2

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta
$ |
```

Aunque deberíamos ya contar con Git instalado, no está de más verificar si realmente contamos con la herramienta en nuestro PC.

git init

```
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta
$ git init
Initialized empty Git repository in E:/EJEMPLO/Carpeta/.git/
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```



Es hora de crear nuestro primer repositorio local, para lo cual basta con ... git init



Para este momento Git debió haber creado una carpeta adicional con el nombre de .git, esto con el fin de almacenar los cambios que vamos haciendo en nuestro repositorio, posiblemente no la veas a simple vista, suele estar oculta

git add .

git add Historia.docx

```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git add .

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```



Luego vamos a empezar a adicionar varios archivos a la vez para el control de cambios, en teoria es almacenar temporalmente los cambios. aquí utilizamos el comando git add .

pero también podríamos utilizar el comando git add nombre y extensión del archivo que deseamos agregar de manera individual.

Nota: en este momento utilizar cualquiera de las dos opciones nos debería entregar el mismo resultado ya que solo contamos con un archivo.

git status



Si todo va bien hasta aquí, podríamos visualizar los archivos que hemos adicionado al repositorio, y deberían estar en verde...

Nota: si se encuentra algún nombre en rojo, este hace referencia a que no se han adicionado al repositorio ni de forma temporal o permanente.

```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git status
On branch master

No commits yet

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

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```

git commit -m "version original"



```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git commit -m "version original"
[master (root-commit) 3c225b7] version original
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Historia.docx

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```

Y en este momento llegamos a un momento crucial, vamos a guardar definitivamente los cambios que se han realizado, este proceso se denomina commit, para ello utilizamos el siguiente comando:  
git commit -m "comentario"  
el comentario es libre, la idea es describir un poco los cambios que se aplicados, o si eres un poco mas avanzado puedes nombrarlo con la versión del archivo.



De nuevo Status, es una buena opción para que revisemos que cambios o que archivos no han sido adicionados al espacio de trabajo, para lo cual los agregas con add para un almacenamiento temporal y los guardas definitivamente con commit.

git status



```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git status
On branch master
nothing to commit, working tree clean

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```

git log

```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git log
commit 3c225b76f59e5a3ddac7156e37b6d0669844ea4 (HEAD -> master)
Author: juliancitofranco <juliancitolindo@ejemplo.com>
Date:   Tue Mar 1 06:40:34 2022 -0500

    version original

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```

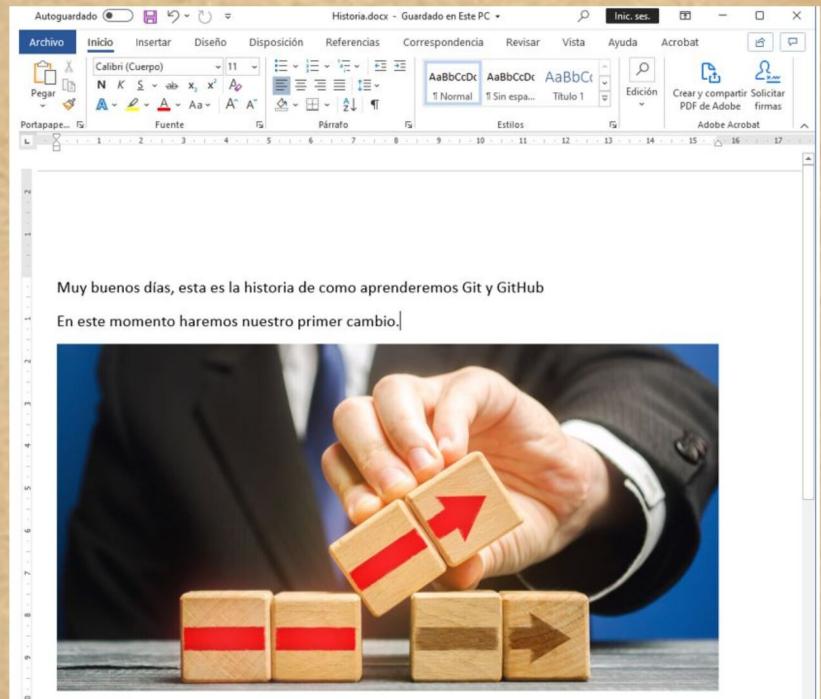


Después de haber realizado nuestro guardado permanente, podemos verificar si se ha almacenado con éxito, para lo cual usaremos el comando  
git log

allí podemos acceder a una lista de los commit que tenemos almacenados, en este momento debería existir tan solo uno. Nos mostrara el autor, la fecha de creación y lo mas importante el código del commit (hash)



En este momento, realizaremos un pequeño cambio a nuestro archivo de Word, colocaremos una nueva frase y una imagen



git status

```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (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: Historia.docx

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

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```



Luego de realizar los cambios en el documento de Word, y de guardarlos, procedemos nuevamente a verificar status en Git, allí deberá aparecernos el archivo en rojo, indicándonos que se ha modificado el archivo y no se han guardado cambios, ni temporal ni definitivamente.



git add . ==> git status ==>  
git commit -m "primer cambio"

```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git status
On branch master
nothing to commit, working tree clean

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```

Es ahora cuando repasaremos lo aprendido, usaremos los comandos, git add . ==> git status ==> git commit -m "primer cambio"

# ADD .

```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git add .

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```

# STATUS

```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Historia.docx

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```

# COMMIT

```
MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git commit -m "primer cambio"
[master be06987] primer cambio
 1 file changed, 0 insertions(+), 0 deletions(-)

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ |
```



Si todo sale bien, aquí podríamos verificar de nuevo el git log y confirmamos si existen nuestros dos commit



Espero que hasta aquí no hayas tenido ningún inconveniente, te haz ganado la segunda rosa de reconocimiento por tu esfuerzo.

git log

```
MINGW64:/e/EJEMPLO/Carpeta
$ git log
commit be069874e9e03c0a08a2de3c2a9737a1ebded60b (HEAD -> master)
Author: juliancitofranco <juliancitolindo@ejemplo.com>
Date: Tue Mar 1 06:53:09 2022 -0500

    primer cambio

commit 3c225b76f59e5a3ddac7156e37b6d0669844eae4
Author: juliancitofranco <juliancitolindo@ejemplo.com>
Date: Tue Mar 1 06:40:34 2022 -0500

    version original

MINGW64:/e/EJEMPLO/Carpeta (master)
$ |
```



```

MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git log
commit be069874e9e03c0a08a2de3c2a9737a1ebded60b (HEAD -> master)
Author: juliancitofranco <juliancitolindo@ejemplo.com>
Date: Tue Mar 1 06:53:09 2022 -0500

primer cambio

commit 3c225b76f59e5a3ddac7156e37b6d0669844eae4
Author: juliancitofranco <juliancitolindo@ejemplo.com>
Date: Tue Mar 1 06:40:34 2022 -0500

    version original

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)

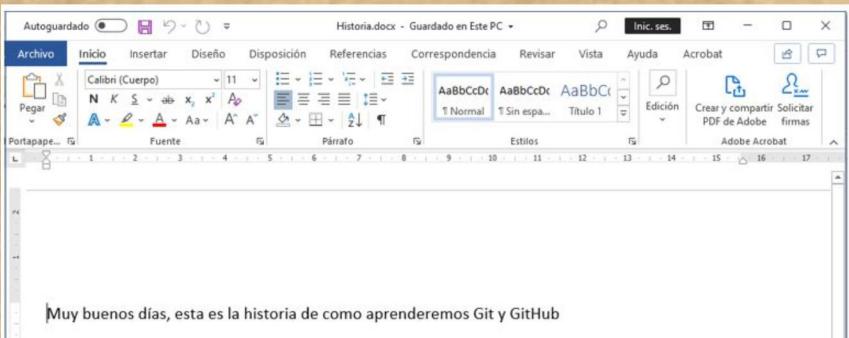
```

Recuerdas que cada commit tiene un código propio, bueno este sirve para poder regresar a el sin inconvenientes ni confusiones, ademas la descripción que le hemos dado servirá para una clara identificación del mismo.



Cuando deseemos regresar a un versión anterior, debemos saber cuales son minimo los 7 primeros digitos del código del commit y luego usaremos el siguiente comando

`git checkout 3c225b76f59e`



```

MINGW64:/e/EJEMPLO/Carpeta
ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta (master)
$ git checkout 3c225b76f59e
Note: switching to '3c225b76f59e'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

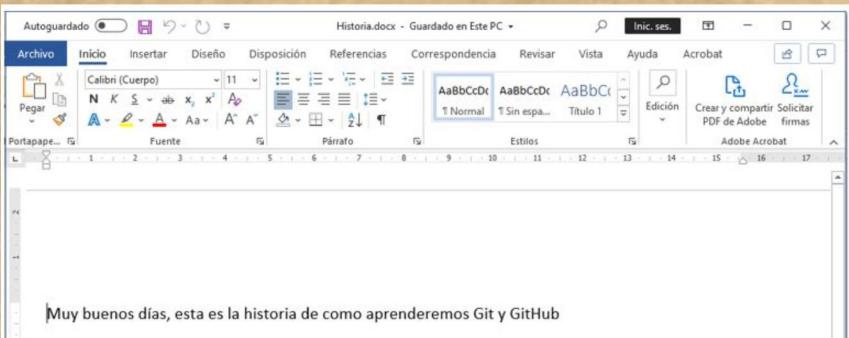
  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 3c225b7 version original

ARMAPRRQDFSD225+SENA@ARMAPRRQDFSD225 MINGW64 /e/EJEMPLO/Carpeta ((3c225b7...))
$ 

```





Si hasta aquí, todo ha resultado exitoso  
te diríamos el archivo en su versión  
original

**Y NOS PODRIAMOS TOMAR**

**!!! UN BREVE DESCANSITO !!!**



Recuerda que entre  
más practiques,  
más pronto  
interiorizaras estos  
conceptos

# ELABORADO POR LOS MEJORES

DIEGO ALZATE

ALEJANDRO GIRALDO

JACOB HOYOS

DIANA RESTREPO

JULIAN FRANCO



# GIT Y GITHUB

---



HOLA, EN ESTE MODULO  
APRENDEREMOS A IMPLEMENTAR  
Y USAR LAS RAMAS VOCALMENTE  
Y EN LA NUBE (GITHUB)



# MODULO 5

# RAMAS

# **IMPLEMENTAR RAMAS HACER CAMBIOS Y FUSIÓN DE ESTAS**



EN ESTE MOMENTO  
ESTAMOS INGRESANDO  
A UN NIVEL INTERMEDIO  
ALTO. DONDE SE  
APLICARAN CONCEPTOS  
DE TRABAJOS  
COLABORATIVOS



En git tenemos la posibilidad de crear entornos o caminos diferentes en cada repositorio, a estos distintos entornos se les llama rama. Cada rama nace con el contenido en base de donde se cree, es decir, si tenemos la rama main y creamos una rama develop esta nueva rama se creará con el contenido que, hasta el momento, tiene main.

Después de crear la rama, esta será una ramificación de la rama de donde se creó, pero en cierto modo será independiente de esta, ya que tendrá sus propios commits, versiones o cambios, cambios que podrán ser integrados a otra rama fusionándose.

Se puede trabajar con ramas desde un repositorio local o remoto.

# RAMAS EN REPOSITORIO LOCAL

Iniciemos un repositorio, yo lo haré en una carpeta con nombre ramas\_git.

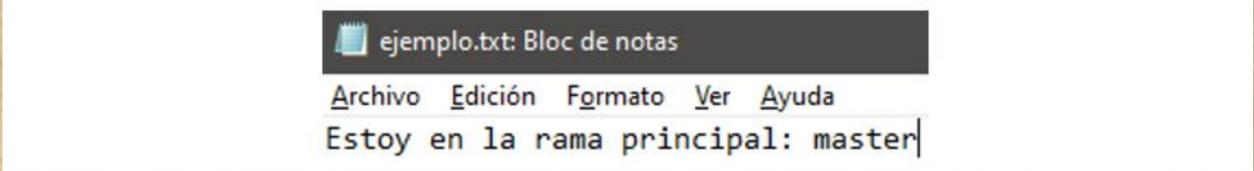
git init

```
PS C:\Users\SENA\ramas_git> git init
Initialized empty Git repository in C:/Users/SENA/ramas_git/.git/
PS C:\Users\SENA\ramas_git> -
```

Cuando creamos el repositorio y hacemos nuestro primer commit, git automáticamente crea una rama principal llamada master.  
Comprobémoslo creando un archivo y guardándolo con git, yo lo haré con un archivo de texto llamado ejemplo.txt.



C:\Users\SENA\ramas_git		Buscar en ramas_git	
Nombre	Fecha de modificación	Tipo	Tamaño
ejemplo.txt	25/02/2022 11:44 a. m.	Documento de te...	0 KB



```
PS C:\Users\SENA\ramas_git> git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ejemplo.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\SENA\ramas.git> git add .
PS C:\Users\SENA\ramas.git> git commit -m "Inicialización del repositorio con el archivo ejemplo.txt"
[master (root-commit) 96f2f02] Inicialización del repositorio con el archivo ejemplo.txt
 1 file changed, 1 insertion(+)
 create mode 100644 ejemplo.txt
PS C:\Users\SENA\ramas.git> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\SENA\ramas.git> -
```



Creamos el archivo y si revisamos el estado de git, veremos que no está en seguimiento. Lo añadiremos para que git lleve un seguimiento de sus versiones y crearemos un commit o guardado de versión.

Ya con este primer commit, git creará la rama git por defecto, podemos verla con **git branch**



```
PS C:\Users\SENA\ramas_git> git branch
* master
PS C:\Users\SENA\ramas_git> -
```

git branch

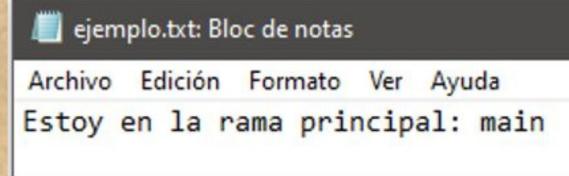
Las ramas podemos renombrarlas, resetearlas, crearlas o eliminarlas. Primero vamos a modificar el nombre de master a main con el comando `git branch -M main`



`git branch -M main`

```
PS C:\Users\SENA\ramas_git> git branch -M main
PS C:\Users\SENA\ramas_git> git branch
* main
PS C:\Users\SENA\ramas_git> -
```

Cambiaré el contenido de nuestro archivo ejemplo.txt para que el nombre de las ramas concuerden.



```
Windows PowerShell
PS C:\Users\SENA\ramas_git> git add .
PS C:\Users\SENA\ramas_git> git commit -m "Cambio de nombre de rama en ejemplo.txt"
[main 529bd4c] Cambio de nombre de rama en ejemplo.txt
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\SENA\ramas_git>
```

Ahora bien, recuerda  
guardar de manera  
permanente, en otras  
palabras haz un  
commit



git add . --> git commit

Para crear una nueva rama y  
poder modificar este archivo  
sin que afecte a la rama main,  
utilizamos el comando **git  
checkout -B  
<nombre\_rama>**:



git checkout -B <nombre\_rama>

```
Windows PowerShell
PS C:\Users\SENA\ramas_git> git branch
* main
PS C:\Users\SENA\ramas_git> git checkout -B develop
Switched to a new branch 'develop'
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git> -
```



Ahora tenemos dos ramas, main y develop. Por ahora, develop es un clon de main, es decir, tiene exactamente los mismos cambios y commits

```
Windows PowerShell
PS C:\Users\SENA\ramas_git> git log
commit 529bd4c589abe155266d9ac20f1aabf77f3332e1 (HEAD -> develop, main)
Author: Jhonatan David Rico Echeverri <xricoxdrumsx@gmail.com>
Date:   Tue Mar 1 06:29:49 2022 -0500

    Cambio de nombre de rama en ejemplo.txt

commit 96f2f02348d2ec2311ae4ff242cf596d8daaf11f
Author:
Date:   Fri Feb 25 11:53:33 2022 -0500

    Inicialización del repositorio con el archivo ejemplo.txt
PS C:\Users\SENA\ramas_git>
```

```
Windows PowerShell
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git>
```



Como vemos, los commits son los mismos. Vamos a hacer un cambio.

Primero que todo tenemos que asegurarnos que estamos en la rama 'develop' con el comando git branch

```
Windows PowerShell
PS C:\Users\SENA\ramas_git> git checkout main
Switched to branch 'main'
PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> git checkout develop
Switched to branch 'develop'
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git>
```



Si llegamos a estar en la rama main podemos desplazarnos entre ramas con el comando `git checkout <nombre_rama>`:



```
ejemplo.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Estoy en la rama de desarrollo: develop
```

```
PS C:\Users\SENA\ramas_git> git status
On branch develop
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:   ejemplo.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\SENA\ramas_git> git add .\ejemplo.txt
PS C:\Users\SENA\ramas_git> git commit -m "Modificando nombre de la rama."
[develop 3e63559] Modificando nombre de la rama.
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\SENA\ramas_git> git status
On branch develop
nothing to commit, working tree clean
PS C:\Users\SENA\ramas_git> _
```

Estando en la rama develop, vamos a hacer un cambio en el archivo ejemplo.txt. Lo agregamos al staging área y le haremos un commit.

Get -Content

Si vemos el contenido del archivo ejemplo.txt en la rama develop, este será diferente al que hay en la main. Comprobémoslo

```
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git> Get-Content .\ejemplo.txt
Estoy en la rama de desarrollo: develop
PS C:\Users\SENA\ramas_git> git checkout main
Switched to branch 'main'
PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> Get-Content .\ejemplo.txt
Estoy en la rama principal: main
PS C:\Users\SENA\ramas_git> -
```



Vemos que el contenido es diferente, ya que develop tiene sus propios commits. Los commits del repositorio los podemos revisar con el comando `git log`

```
PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> git checkout develop
Switched to branch 'develop'
PS C:\Users\SENA\ramas_git> git branch
* develop
  main
PS C:\Users\SENA\ramas_git> git log
commit 3e6355933169e9ef9f10531207dfdbb0d28034ec (HEAD -> develop)
Author: Jhonatan David Rico Echeverri <xricoxdrumsx@gmail.com>
Date:   Tue Mar 1 07:14:55 2022 -0500

    Modificando nombre de la rama.

commit 529bd4c589abe155266d9ac20f1aabf77f3332e1 (main)
Author:
Date:   Tue Mar 1 06:29:49 2022 -0500

    Cambio de nombre de rama en ejemplo.txt

commit 96f2f02348d2ec2311ae4ff242cf596d8daaf11f
Author: Jhonatan David Rico Echeverri <xricoxdrumsx@gmail.com>
Date:   Fri Feb 25 11:53:33 2022 -0500

    Inicialización del repositorio con el archivo ejemplo.txt
PS C:\Users\SENA\ramas_git> -
```

git log

Te haz ganado una rosa por tu esfuerzo, si todo nos ha salido bien hasta aquí ... hemos aprendido a:

- Crear ramas en git.
- Cambiar de nombre a una rama.
- Cambiar entre ramas.
- Ver los commits de las ramas.

Ahora supongamos que en develop tenemos cambios importantes que queremos pasar a la rama main ¿Cómo hacemos esto?, para esto utilizamos la fusión de ramas, la cual consiste en juntar cambios de una rama con otra y, en caso de que haya conflictos, decidir qué cambios aceptar en la mezcla.



Para hacer una fusión o merge, nos ubicamos en la rama que queremos que se fusioné con otra, en este caso, main.

```
PS C:\Users\SENA\ramas_git> git checkout main
Switched to branch 'main'
PS C:\Users\SENA\ramas_git>
```



git merge <nombre\_rama>

```
PS C:\Users\SENA\ramas_git> git merge develop
Updating 529bd4c..3e63559
Fast-forward
  ejemplo.txt | 2 +-
  1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\SENA\ramas_git>
```

Ahora la rama main tiene los cambios de la rama develop

Veamos el estado de la rama  
git status

```
PS C:\Users\SENA\ramas_git> git status
On branch main
nothing to commit, working tree clean
PS C:\Users\SENA\ramas_git> _
```



git merge develop



Todo está limpio, no hay cambios sin guardar, pero si miramos el contenido de ejemplo.txt ...

```
PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> Get-Content .\ejemplo.txt
Estoy en la rama de desarrollo: develop
PS C:\Users\SENA\ramas_git>
```

git branch



Así es, los cambios que llevábamos en develop ahora están también en main y en los logs las dos ramas aparecen en el mismo nivel de commit, en el último.

```
PS C:\Users\SENA\ramas_git> git branch
  develop
* main
PS C:\Users\SENA\ramas_git> Get-Content .\ejemplo.txt
Estoy en la rama de desarrollo: develop
PS C:\Users\SENA\ramas_git>
```

Por tu esfuerzo has ganado una rosa, si todo va bien hasta aquí lo haz hecho muy bien

```
git log
```

```
PS C:\Users\SENA\ramas_git> git log
commit 3e6355933169e9ef9f10531207dfdbb0d28034ec (HEAD -> main, develop)
Author:
Date: Tue Mar 1 07:14:55 2022 -0500

    Modificando nombre de la rama.

commit 529bd4c589abe155266d9ac20f1aabf77f3332e1
Author:
Date: Tue Mar 1 06:29:49 2022 -0500

    Cambio de nombre de rama en ejemplo.txt

commit 96f2f02348d2ec2311ae4ff242cf596d8daaf11f
Author: Jhonatan David Rico Echeverri <xricoxdrumsx@gmail.com>
Date: Fri Feb 25 11:53:33 2022 -0500

    Inicialización del repositorio con el archivo ejemplo.txt
PS C:\Users\SENA\ramas_git> -
```

Ahora bien, si no necesitamos más la rama de develop, podemos eliminarla con el comando git branch -D <nombre\_rama>:

```
PS C:\Users\SENA\ramas_git> git branch -D develop
Deleted branch develop (was 3e63559).
PS C:\Users\SENA\ramas_git> git branch
* main
PS C:\Users\SENA\ramas_git> -
```



```
git branch -D develop
```

Debemos tener en cuenta que:

- Las ramas no se pueden eliminar si hay cambios en el stagin area o si hay cambios no añadidos o no removidos.
- La eliminación de una rama NO SOLICITA CONFIRMACIÓN por lo que hay que tener cuidado.

Las ramas se pueden subir a un repositorio remoto, en caso de que exista con el comando git push <url o alias del repositorio remoto> <rama>:

## RAMAS EN REPOSITORIO REMOTO



Ya vimos cómo manejar ramas en un repositorio local, ahora miraremos cómo trabajar con ramas en un repositorio remoto en Github.

En este caso, tomaré un repositorio remoto ya creado y con un archivo de configuración ya que hasta este punto debes de saber cómo crear tu cuenta en GitHub, como crear un repositorio remoto y cómo iniciararlo.

Te felicito por tu compromiso hasta este punto. Lo haz hecho muy bien

# GITHUB

A screenshot of a GitHub repository page for 'xAGH / Cloud-formation'. The page shows a single branch named 'main'. A red arrow points to the 'main' branch button. The interface includes standard GitHub navigation tabs like Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below the main branch, there is a file named 'DeployExample.yaml' under the 'cloudformation' folder. A small message at the bottom encourages adding a README.



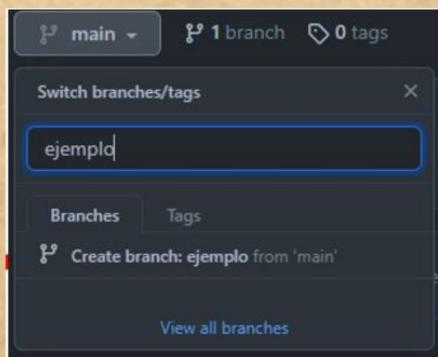
Este es nuestro repositorio



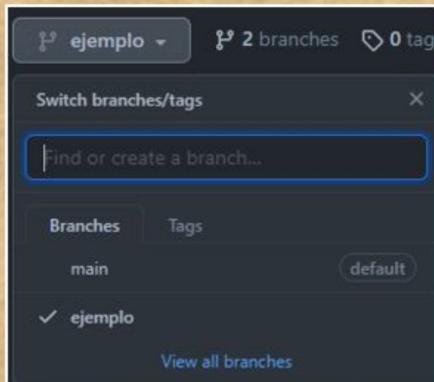
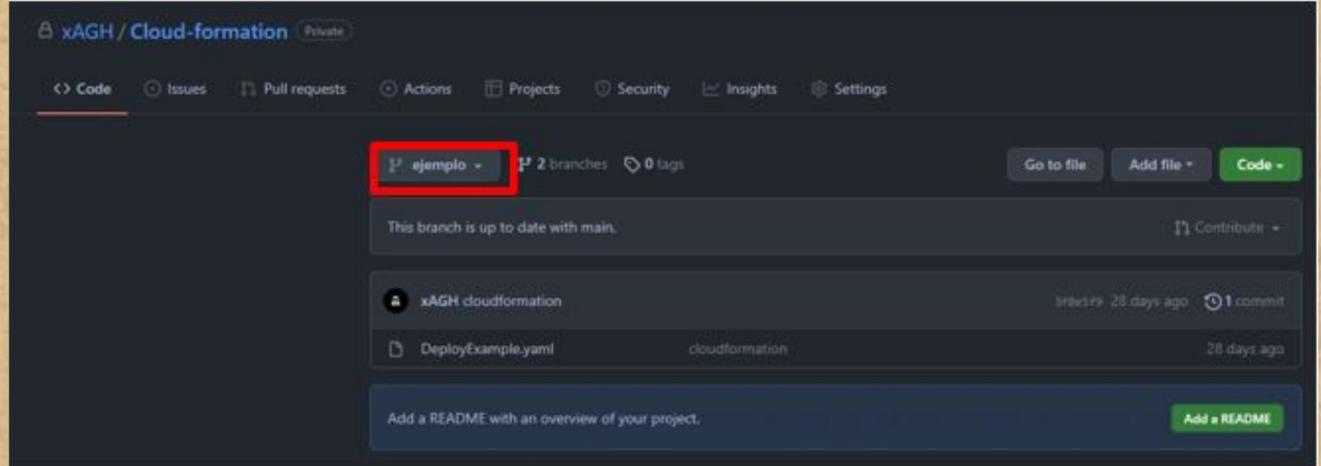
Si desplegamos  
las ramas,  
veremos sólo  
una

A screenshot of the GitHub branch selector. It shows the 'main' branch selected. A modal window titled 'Switch branches/tags' is open, with a search bar containing the placeholder 'Find or create a branch...'. Below the search bar, there are tabs for 'Branches' and 'Tags', with 'Branches' currently selected. Under the 'Branches' tab, only the 'main' branch is listed. A 'default' button is visible next to the branch names. At the bottom of the modal, there is a link 'View all branches'.

Para crear una rama desde el repositorio remoto, buscamos en la barra de búsqueda el nombre que queramos poner a nuestra rama, en mi caso ‘ejemplo’. Si esta rama no existe en el repositorio remoto, git la creará, si existe, simplemente la busca.



En este caso crearemos la rama ejemplo



Ya la rama está creada. Si desplegamos las ramas, veremos las dos



Solicitar revisiones y autorizaciones: En grupos de trabajo grandes, es necesario que los commits subidos al repositorio remoto sean revisados y aprobados para evitar problemas en el despliegue o introducir bugs en el código, aquí vamos a simular una de estas situaciones, vamos a preparar nuestro entorno



Ahora podemos hacer cambios en los archivos y estos quedarán guardados exclusivamente en su rama hasta que se fusionen

### Repositorio local

```
~/ramas_git > git init
Inicializado repositorio Git vacío en /home/alejo/ramas_git/.git/
~/ramas_git master > echo "Desde la rama main" > ejemplo.txt
~/ramas_git master ?1 > git add .
~/ramas_git master +1 > git commit -m "Inicio repo con ejemplo.txt"
[master (commit-raíz) 5c6ee54] Inicio repo con ejemplo.txt
 1 file changed, 1 insertion(+)
 create mode 100644 ejemplo.txt
~/ramas_git master > git branch -M main
~/ramas_git main > git checkout -B develop
Cambiado a nueva rama 'develop'
~/ramas_git develop > echo "Modificado desde develop" > ejemplo.txt
~/ramas_git develop !1 > git add .
~/ramas_git develop +1 > git commit -m "Modificacion en develop de ejemplo.txt"
[develop 605fd5b] Modificacion en develop de ejemplo.txt
 1 file changed, 1 insertion(+), 1 deletion(-)
~/ramas_git develop > |
```

Ahora creamos el repositorio remoto para enlazarlo con el local

Alejo Giraldo  
XAGH  
Globo redondeado

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

**Owner \***      **Repository name \***

xAGH / ramas\_git ✓

Great repository name! ramas\_git is available. Create. Need inspiration? How about verbose-octo-winner?

**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:**

Skip this step if you're importing an existing repository.

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

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more](#).

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more](#).

**Create repository**



### Conexión a repositorio remoto

```

~/ramas_git develop > git checkout main
Cambiado a rama 'main'
~/ramas_git main > git remote add origin https://github.com/xAGH/ramas_git.git
~/ramas_git main > git push
fatal: La rama actual main no tiene una rama upstream.
Para realizar un push de la rama actual y configurar el remoto como upstream, use

    git push --set-upstream origin main

~/ramas_git main > git push origin main
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 243 bytes | 243.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/xAGH/ramas_git.git
 * [new branch]      main -> main
~/ramas_git main > git checkout develop
Cambiado a rama 'develop'
~/ramas_git develop > git push origin develop
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Escribiendo objetos: 100% (3/3), 285 bytes | 285.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:     https://github.com/xAGH/ramas_git/pull/new/develop
remote:
To https://github.com/xAGH/ramas_git.git
 * [new branch]      develop -> develop
~/ramas_git develop >

```

git remote add origin <url>  
git push origin main

Ahora en nuestro repo remoto tenemos

Como vemos en nuestra rama main nos aparece una alerta que nos dice que en develop hay cambios que no están en la main, creamos otro archivo en develop para visualizar mejor los cambios

xAGH / ramas\_git Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

develop had recent pushes less than a minute ago

Compare & pull request

main 2 branches 0 tags

Go to file Add file Code

xAGH Inicio repo con ejemplo.txt 5c8ee54 11 minutes ago 1 commit

ejemplo.txt Inicio repo con ejemplo.txt 11 minutes ago

Add a README

echo "otro archivo mas" > Otro.txt

```
~/ramas_git develop > echo "Otro archivo mas" > Otro.txt
~/ramas_git develop ?1 > git add .
~/ramas_git develop +1 > git commit -m "Se agrega otro archivo mas"
[develop 4181678] Se agrega otro archivo mas
 1 file changed, 1 insertion(+)
 create mode 100644 Otro.txt
~/ramas_git develop > git push origin develop
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 301 bytes | 301.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/xAGH/ramas_git.git
 605fd5b..4181678  develop -> develop
~/ramas_git develop > |
```

A screenshot of a GitHub profile page for Alejo Giraldo (xAGH). The profile picture is a stylized yellow and black logo. The page shows three repositories: 'SENA' (Public, Python, updated 7 hours ago), 'Cloud-formation' (Private, updated 21 hours ago), and 'ContabinniApp' (Public, updated 18 days ago). There are buttons to 'Star' each repository. Below the repositories, it says 'At 5 followers · 3 following' and a 'Edit profile' button.

### Create a new repository

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

**Owner \***      **Repository name \***

xAGH /  ✓

Great repository name! ramas\_git is available. Need inspiration? How about [verbose-octo-winner](#)?

**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:**  
Skip this step if you're importing an existing repository.

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

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more](#).

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more](#).

**Create repository**



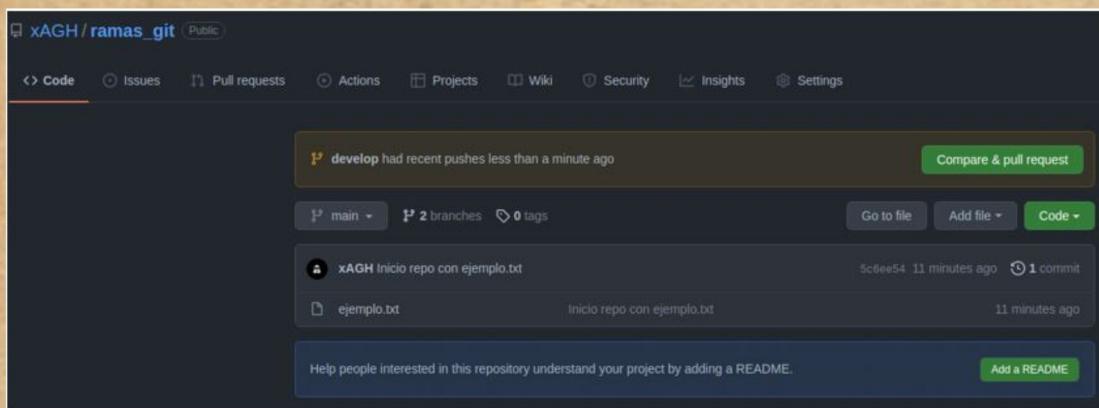
Ahora creamos el repositorio remoto para enlazarlo con el local

## Conexión a repositorio remoto

```
~/ramas_git develop > git checkout main
Cambiado a rama 'main'
~/ramas_git main > git remote add origin https://github.com/xAGH/ramas_git.git
~/ramas_git main > git push
fatal: La rama actual main no tiene una rama upstream.
Para realizar un push de la rama actual y configurar el remoto como upstream, use

    git push --set-upstream origin main

~/ramas_git main > git push origin main
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 243 bytes | 243.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/xAGH/ramas_git.git
 * [new branch]      main -> main
~/ramas_git main > git checkout develop
Cambiado a rama 'develop'
~/ramas_git develop > git push origin develop
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Escribiendo objetos: 100% (3/3), 285 bytes | 285.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:     https://github.com/xAGH/ramas_git/pull/new/develop
remote:
To https://github.com/xAGH/ramas_git.git
 * [new branch]      develop -> develop
~/ramas_git develop > |
```

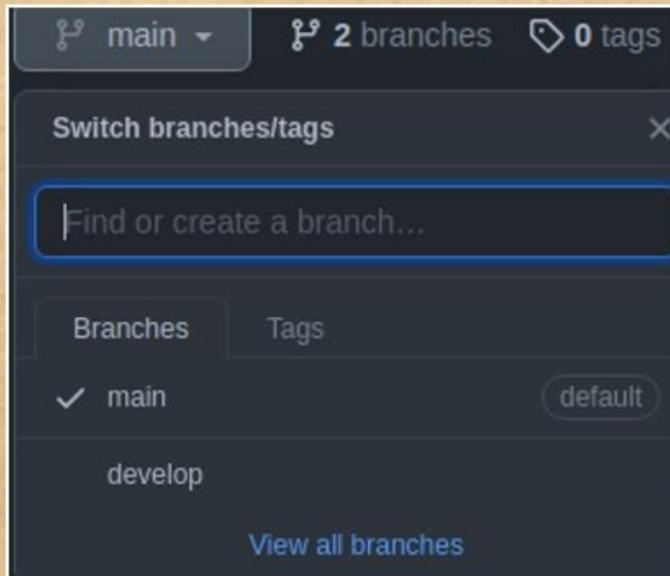


The screenshot shows a GitHub repository page for 'xAGH / ramas\_git'. The page includes navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area displays two branches: 'main' (selected) and 'develop'. It also shows 0 tags. A commit history is listed, with the most recent one being 'xAGH Inicio repo con ejemplo.txt' at 5c6ee54, 11 minutes ago, which has 1 commit and was made by 'ejemplo.txt' at 11 minutes ago. A message at the bottom encourages adding a README.

Ahora en  
nuestro repo  
remoto tenemos



Como vemos en nuestra rama main nos aparece una alerta que nos dice que en develop hay cambios que no están en la main, creamos otro archivo en develop para visualizar mejor los cambios



```
~/ramas_git develop > echo "Otro archivo mas" > Otro.txt
~/ramas_git develop ?1 > git add .
~/ramas_git develop +1 > git commit -m "Se agrega otro archivo mas"
[develop 4181678] Se agrega otro archivo mas
 1 file changed, 1 insertion(+)
  create mode 100644 Otro.txt
~/ramas_git develop > git push origin develop
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 301 bytes | 301.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/xAGH/ramas_git.git
  605fd5b..4181678  develop -> develop
~/ramas_git develop > |
```



Si todo nos ha ido bien hasta aqui ... Esto nos aparece en el repositorio remoto y te mereces una rosa por tu buen trabajo

⚡ develop had recent pushes 1 minute ago

Compare & pull request

main ▾ 2 branches 0 tags

Go to file Add file ▾ Code ▾

xAGH Inicio repo con ejemplo.txt 5c6ee54 14 minutes ago 1 commit

ejemplo.txt Inicio repo con ejemplo.txt 14 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

Un pull request es una solicitud que hace develop hacia main para integrar los cambios, como yo soy el dueño y único colaborador del repositorio podré hacer y aceptar el pull request por mi cuenta



Vamos a ver cómo funciona un pull

A screenshot of a GitHub repository page. At the top, there's a message from a user named 'develop' stating they had recent pushes 8 minutes ago. To the right of this message is a green button with the text 'Compare & pull request', which is highlighted with a red rectangular border. Below this, there are navigation links for 'main', '2 branches', '0 tags', and buttons for 'Go to file', 'Add file', and 'Code'. The main content area shows a commit from 'xAGH' titled 'Inicio repo con ejemplo.txt' made 21 minutes ago at commit hash '5c6ee54'. Below the commit is a file listing for 'ejemplo.txt' with the same details. At the bottom of the page, there's a call-to-action bar encouraging users to add a README file.

develop had recent pushes 8 minutes ago

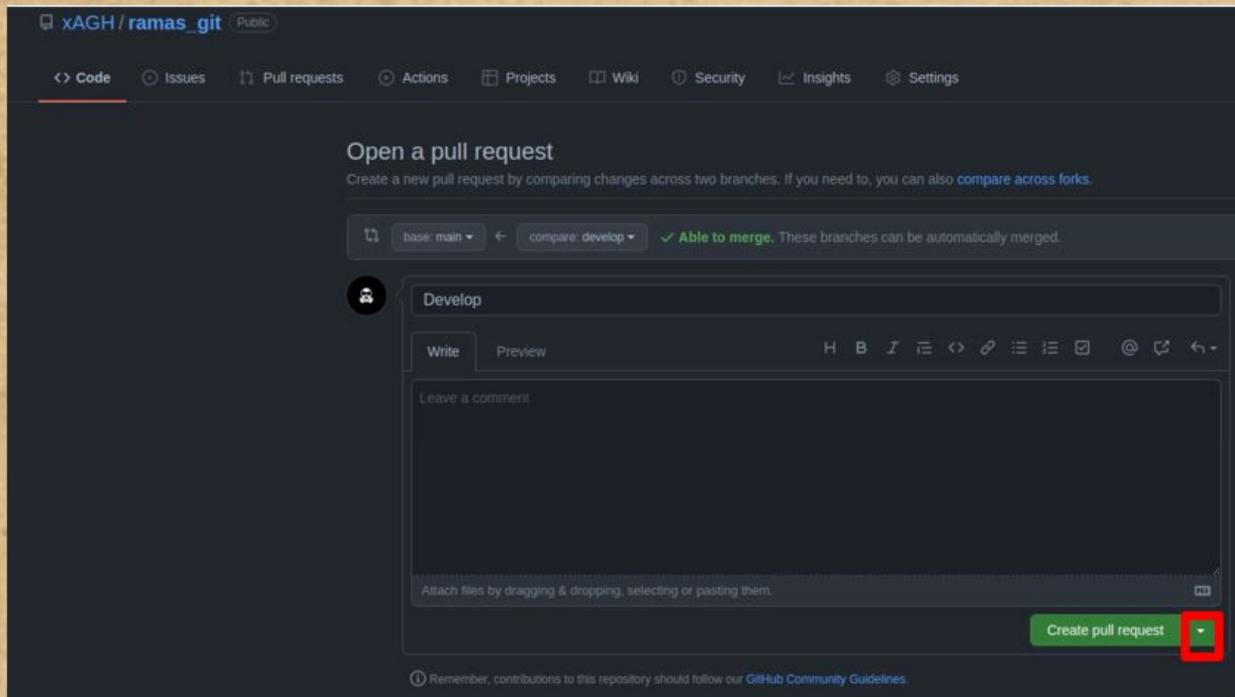
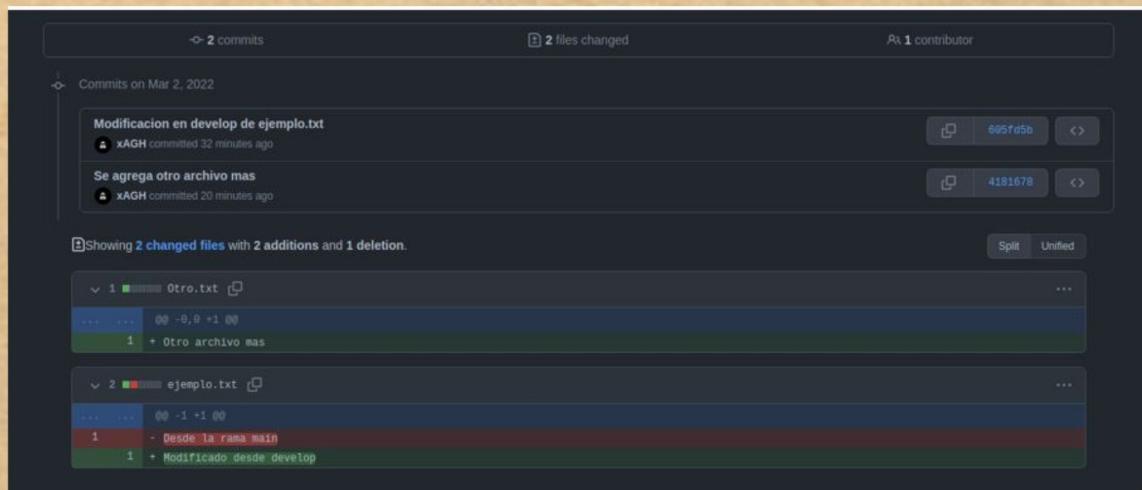
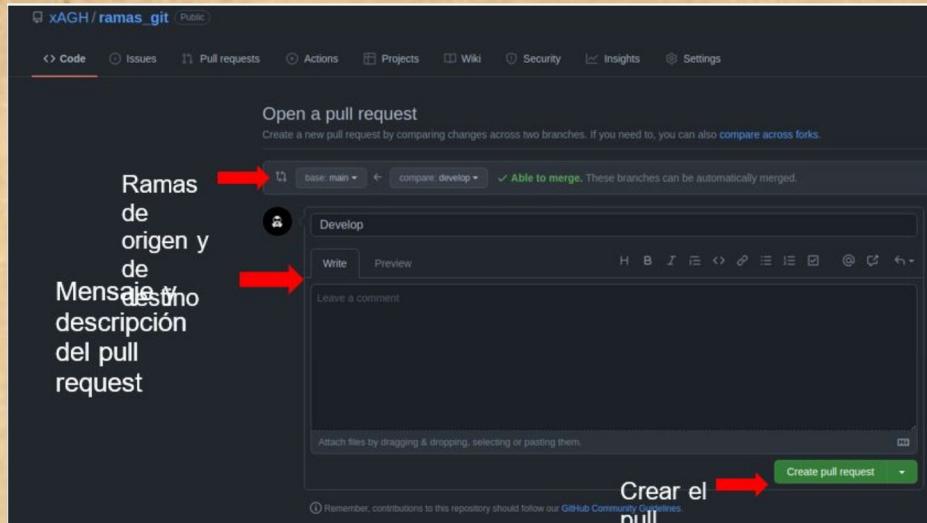
Compare & pull request

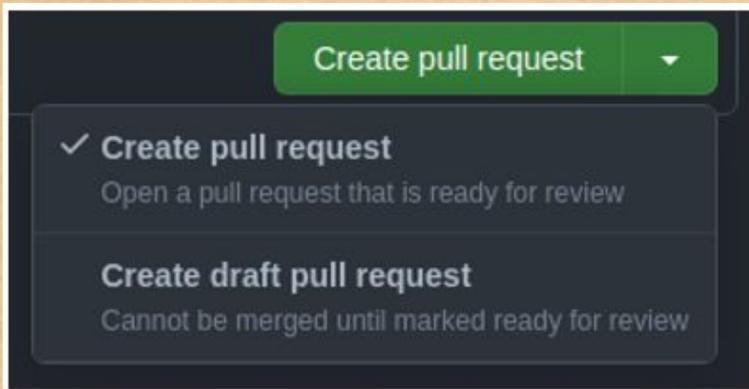
main 2 branches 0 tags Go to file Add file Code

xAGH Inicio repo con ejemplo.txt 5c6ee54 21 minutes ago 1 commit

ejemplo.txt Inicio repo con ejemplo.txt 21 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README





Create pull request: Crea un pull request listo para ser revisado y aprobado.

Create draft pull request: Crea un pull request que debe ser marcado como apropiado para ser revisado y aprobado.

Ahora revisemos el pull request



Create pull request

A screenshot of the 'Open a pull request' interface. It shows a dark-themed form with the title 'Open a pull request'. At the top, it says 'Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.' Below this, there are dropdown menus for 'base: main' and 'compare: develop'. A green checkmark indicates 'Able to merge. These branches can be automatically merged.' The main area contains a text input field labeled 'Ejemplo de pull request' with the placeholder text 'De la rama develop se tomará el cambio en el archivo ejemplo.txt y se añadirá Otro.txt'. There are 'Write' and 'Preview' tabs above the text area. At the bottom, there is a note about attaching files and a red-bordered 'Create pull request' button.

Los pull request aparecerán en esta sección

Ejemplo de pull request #1

xAGH wants to merge 2 commits into `main` from `develop`

Conversation Commits Checks Files changed

xAGH commented 1 minute ago  
De la rama develop se tomará el cambio en el archivo ejemplo.txt y se añadirá Otro.txt

xAGH added 2 commits 43 minutes ago  
Modificación en develop de ejemplo.txt  
Se agrega otro archivo mas

Add more commits by pushing to the `develop` branch on `xAGH/ramas.git`.

This branch has no conflicts with the base branch  
Merging can be performed automatically.

Merge pull request or view command line instructions.

Write Preview  
Leave a comment **Globo de exclamación alternativo**

Attach files by dragging & dropping, selecting or pasting them

**Cerrar el pull request, no aceptarlo**

Vamos a aceptarlo



Ejemplo de pull request #1

xAGH wants to merge 2 commits into `main` from `develop`

Conversation Commits Checks Files changed

xAGH commented 1 minute ago  
De la rama develop se tomará el cambio en el archivo ejemplo.txt y se añadirá Otro.txt

xAGH added 2 commits 43 minutes ago  
Modificación en develop de ejemplo.txt  
Se agrega otro archivo mas

Add more commits by pushing to the `develop` branch on `xAGH/ramas.git`.

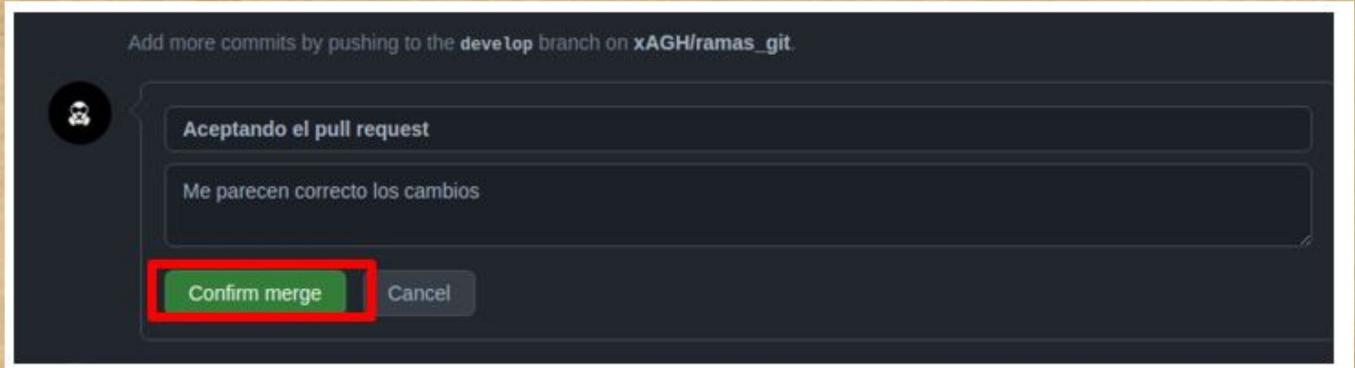
This branch has no conflicts with the base branch  
Merging can be performed automatically.

Merge pull request or view command line instructions.

Write Preview  
**Globo irregular alternativo**

Leave a comment

Attach files by dragging & dropping, selecting or pasting them



Screenshot of a GitHub Pull Request page titled "Ejemplo de pull request #1". The tab bar at the top shows "Code", "Issues", "Pull requests" (which is active), "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings".

The main content area displays a message from user "xAGH" stating: "De la rama develop se tomará el cambio en el archivo ejemplo.txt y se añade Otro.txt". Below this, a commit history shows two commits from "xAGH": "Modificación en develop de ejemplo.txt" and "Se agrega otro archivo mas". A message from "xAGH" indicates the merge: "xAGH merged commit e758b61 into main now".

A success message at the bottom states: "Pull request successfully merged and closed. You're all set—the develop branch can be safely deleted." There is a "Delete branch" button next to it.

At the bottom, there is a comment section with a "Leave a comment" input field and a "Comment" button.



El pull request ha sido aceptado y las ramas combinado

Screenshot of a GitHub repository overview page. The top navigation bar includes "Code" (active), "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings".

The repository stats show "main", "2 branches", and "0 tags". Action buttons include "Go to file", "Add file", and "Code".

A recent activity log entry from "xAGH" says: "Aceptando el pull request" (Accepting the pull request) at "e758b61 1 minute ago" with "4 commits". The commit message "Me parecen correcto los cambios" is shown.

The commit log lists two commits:

- "Otro.txt" - "Se agrega otro archivo mas" - "40 minutes ago"
- "ejemplo.txt" - "Modificación en develop de ejemplo.txt" - "1 hour ago"

At the bottom, a call-to-action button says "Add a README".



Felicitaciones ... Ya con esto hemos aprendido a hacer pull request, revisarlas y aceptarlas o rechazarlas

# ELABORADO POR LOS MEJORES

DIEGO ALZATE

ALEJANDRO GIRALDO

JACOB HOYOS

DIANA RESTREPO

JULIAN FRANCO

