

“Año de la recuperación y consolidación de la economía peruana”

“Universidad Peruana los Andes”

Facultad de Ingeniería

Escuela profesional de Ingeniería de Sistemas y Computación



Semana 1:

.....

Git, Que es y su utilidad

.....

Curso : Desarrollo de Aplicativos I

Docente: Mg.Ing.Bejarano Raul Fernandez

Estudiante: Cruz Quispe Miguel Angel

Ciclo : IVC

Sección: A1

HUANCAYO-PERU 2025

Índice

Que es Git	3
Funcionamiento	3
Paso 1: Descargamos la aplicación	3
Paso 2: Ingresamos al programa	4
Paso 3: Creamos nuestro repositorio	4
Paso 4: Manejamos la primera versión de un programa	5

Que es Git

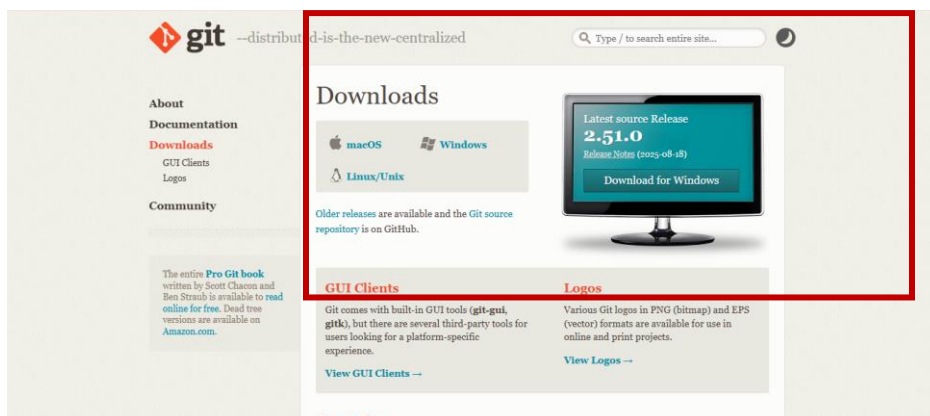
Git es un sistema de control de versiones distribuido, esto significa que sirve para registrar los cambios que haces en tus archivos (código, documentos, etc.) y guardar su historial para que puedas regresar a un código anterior cuando sea necesario, trabajar en equipo sin perjudicar el trabajo de los demás, tener copias de seguridad en caso de cambios drásticos y poder guardar información en la plataforma Github.

Funcionamiento

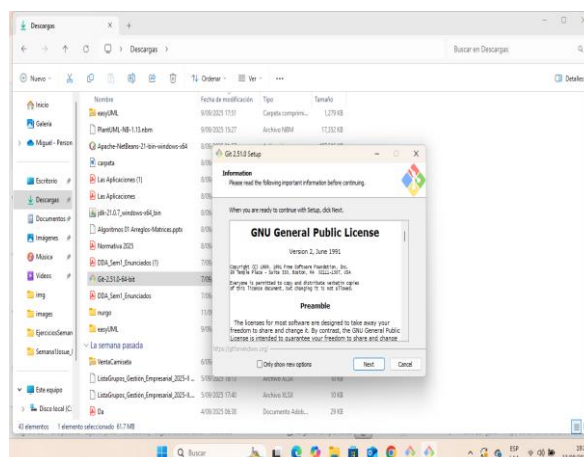
Git tiene un repositorio interno que controla, ahí guarda nuestros archivos y el historial de cambios, estos funcionan en formato de cola así puedes cargar de manera consecutiva y puedes acceder a cualquier archivo del historial en cualquier momento, ahora mostraremos como se realiza este historial y manejo de versiones con los siguientes pasos.

Paso 1: Descargamos la aplicación

Ingresamos a la página web oficial de Git <https://git-scm.com/downloads> y seleccionamos la descarga según nuestro sistema operativo.



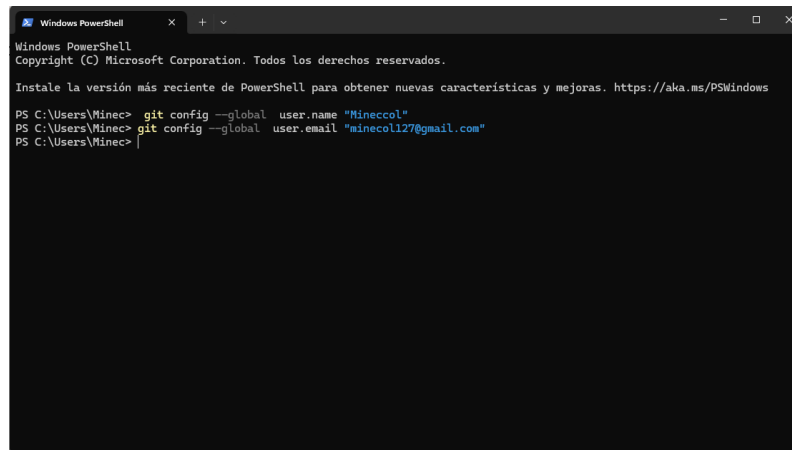
Luego ejecutaremos el sistema de descarga, seguiremos el proceso seleccionando el lugar donde se guardara los archivos internos clásico de manera local.



Paso 2: Ingresamos al programa

ingresamos a la terminal para poder colocar nuestro **username** y **email**, esto como clave de acceso a el repositorio que vamos a crear, ingresamos nuestros datos con el siguiente formato:

`git config --global user.name "Tu nombre"` y `git config --global user.email "Tu email"`



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\Minec> git config --global user.name "Mineccol"
PS C:\Users\Minec> git config --global user.email "mineccol27@gmail.com"
PS C:\Users\Minec> |
```

Paso 3: Creamos nuestro repositorio

Creamos una carpeta y le damos un nombre para saber en donde se guarda, notación simple en vez de espacios se colocan subguiones para ello nos vamos a la terminal de git para crearlo.

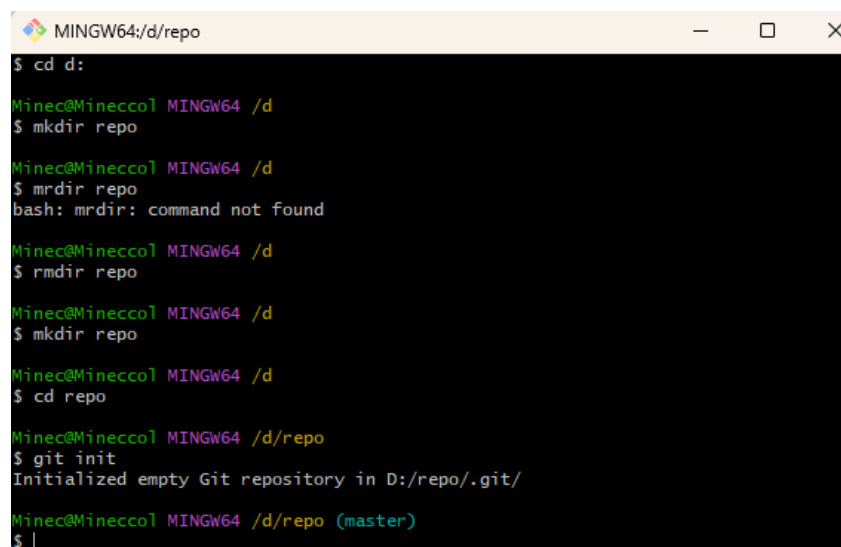
Notación :

Ubicación : `cd d: , /.. , NombreCarpeta.`

Crear Carpeta: `mkdir (nombre de la carpeta)`

Borrar Carpeta: `rmdir (nombre de la carpeta)`

Iniciar repositorio: `git init`



```
MINGW64/d/repo
$ cd d:
Minec@Mineccol MINGW64 /d
$ mkdir repo
Minec@Mineccol MINGW64 /d
$ mrdird repo
bash: mrdird: command not found
Minec@Mineccol MINGW64 /d
$ rmdir repo
Minec@Mineccol MINGW64 /d
$ mkdir repo
Minec@Mineccol MINGW64 /d
$ cd repo
Minec@Mineccol MINGW64 /d/repo
$ git init
Initialized empty Git repository in D:/repo/.git/
Minec@Mineccol MINGW64 /d/repo (master)
$ |
```

Paso 4: Manejamos la primera versión de un programa

Luego crearemos la primera versión de algo en este caso usaremos un archivo txt para probar las funciones básicas de este

Notación:

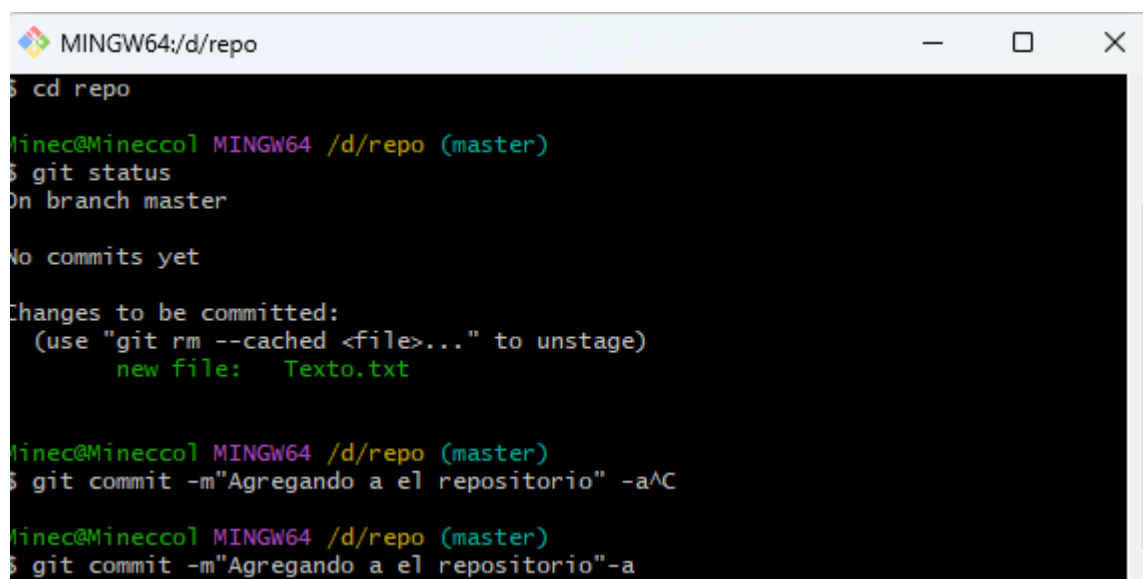
Verificación: ls (para ver que archivos están dentro(menos .git))

Preparar archivo: git add (nombre/extensión)

Eliminar de la zona de preparación: rm --cached (nombre/extensión)

Verificar archivos que están en área de preparación: git status.

Cargar a el repositorio local :git commit -m "Comentario" -a

A screenshot of a terminal window titled 'MINGW64:/d/repo'. The terminal shows the following commands and output:

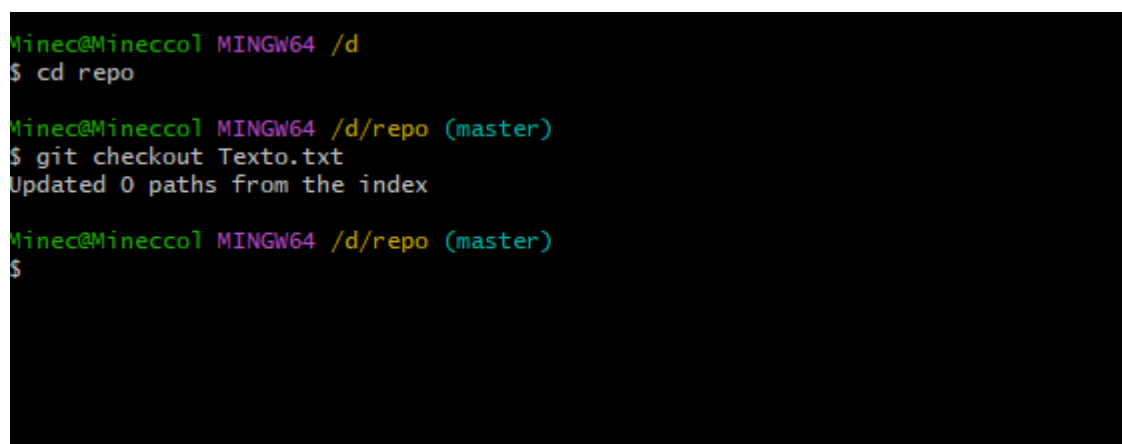
```
$ cd repo
minec@Minecco1 MINGW64 /d/repo (master)
$ git status
On branch master

No commits yet

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

minec@Minecco1 MINGW64 /d/repo (master)
$ git commit -m"Agregando a el repositorio" -a^C
minec@Minecco1 MINGW64 /d/repo (master)
$ git commit -m"Agregando a el repositorio"-a
```

Ahora cabe aclarar que con ello creamos la primera versión del Texto.txt y para regresar a ese "Punto de guardado" se coloca el siguiente comando, el archivo se modifica a ese punto de guardado: **git checkout (nombre/extension).**

A screenshot of a terminal window showing the following commands and output:

```
minec@Minecco1 MINGW64 /d
$ cd repo

minec@Minecco1 MINGW64 /d/repo (master)
$ git checkout Texto.txt
Updated 0 paths from the index

minec@Minecco1 MINGW64 /d/repo (master)
$
```