



Control de versiones con Git (CLI).

¿Que es una versión?

Una versión se entiende como el estado actual de desarrollo de un archivo o programa, a la cual se le asigna un nombre o números.

¿Que es un controlador de versiones?

Un controlador de versiones es un sistema que registra y almacena cada cambio realizado en un archivo o en una serie de estos, desde su creación hasta su posterior eliminación, permitiendo volver a una versión anterior e incluso recuperar archivos borrados previamente registrados.

Tipos de control de versiones.¹

Principalmente se utilizan 3 tipos de control de versiones, **local**, **centralizado** y **distribuido**.

- **Control de versiones local:** Es el tipo de controlador utilizado por la mayoría de las personas, tanto en el hogar como en el trabajo, consiste básicamente en copiar archivos y/o carpetas, hacia otros directorios, sin embargo no es un método efectivo, ya que es propenso a producir errores, como por ejemplo, eliminar archivos, sobrescribir estos mismos o simplemente olvidar la ruta donde se guardaron. (vea Figura 1)

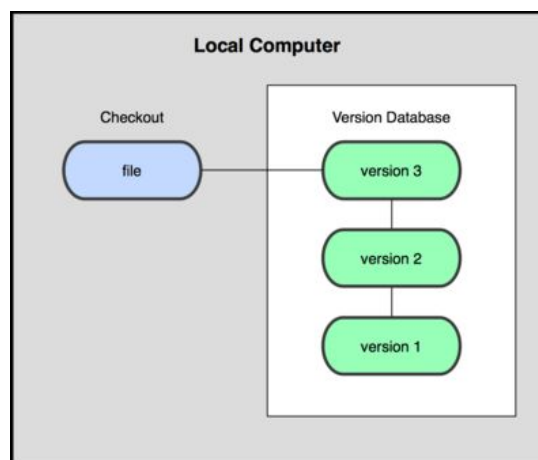


Figura 1. Diagrama de control de versiones local.

¹ Git - Acerca del control de versiones, git book, , consultado el (12/05/2019).
(<https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>).

- Control de versiones centralizado:** A la hora de realizar o elaborar grandes proyectos, debemos colaborar con un sinfín de personas, no trabajamos solos, es por esto que bajo esta necesidad se crean los controladores de versiones centralizados, en los cuales se puede saber bajos ciertos criterios que está desarrollando, editando o produciendo nuestro colaborador. Estos archivos son almacenados en un servidor central, desde el cual personas con acceso pueden descargar versiones o actualizar estas mismas, para posterior uso de cualquier participante del proyecto.(vea Figura 2)

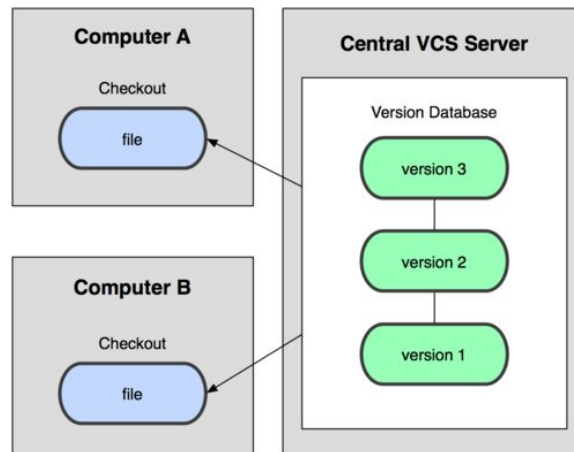


Figura 2. Diagrama de control de versiones centralizado.

- Control de versiones distribuido:** Similar al control de versiones centralizado, pero en este, cada colaborador copia completamente el proyecto en su dispositivo, lo cual permite en caso de perdida de informacion o caída del servidor restaurar todos los ficheros del proyecto.(vea Figura 3)

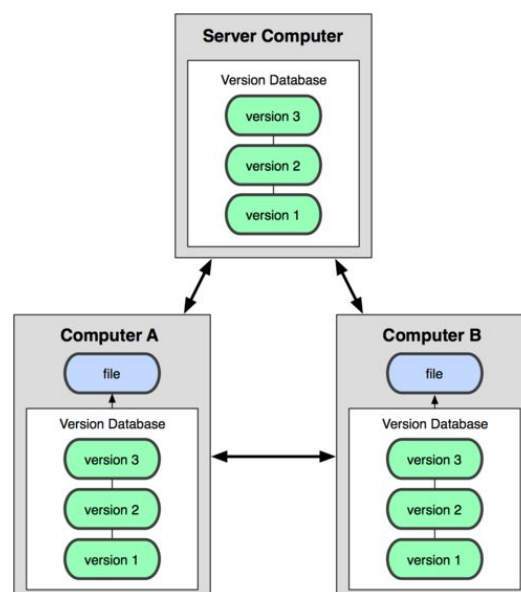


Figura 3. Diagrama de control de versiones distribuido.

Primeros pasos.

Para comenzar comenzaremos instalando Git en nuestro respectivo sistema operativo los ejemplos y prácticas se desarrollarán en windows y linux, cabe mencionar que este es un tutorial básico en entorno CLI (interfaz de línea de comandos), al final de este mismo veremos los distintos GUI (interfaz de usuario gráfica), que nos permitan una utilización visual de nuestro controlador de versiones.

Instalar Git².

- Git-Windows
- Git-Mac
- Git-Linux
 - \$ apt-get install git
 - \$ yum install git-core

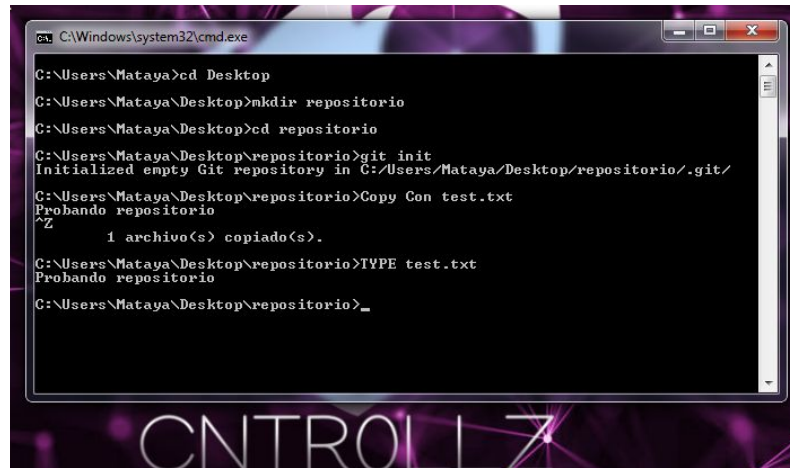
Primer Repositorio local.

Para comenzar crearemos un repositorio local, luego iniciaremos el controlador de versiones git, añadiremos y editaremos archivos, para terminar volviendo a una versión anterior recuperando el trabajo realizado.

- Abrimos nuestro intérprete de comandos y nos posicionamos en un directorio, en esta ocasión utilizaremos el escritorio.
 - cd Desktop
- Creamos nuestra carpeta para utilizar como repositorio.
 - mkdir repositorio
- Ingresamos en nuestro repositorio e iniciamos nuestro controlador de versiones git.
 - cd repositorio
 - git init
- Crearemos nuestro primer archivo, en este paso diferenciaremos comandos para Windows y comandos para Linux.
 - Windows
 - COPY CON test.txt
 - Probando repositorio
 - Ctrl + Z
 - Linux
 - cat > test.txt
 - Probando repositorio
 - Ctrl + D
- Para corroborar leeremos el contenido del archivo (opcional).

² Git - Instalando Git, git book, , consultado el (12/05/2019). (<https://git-scm.com/book/es/v1/Empezando-Instalando-Git>).

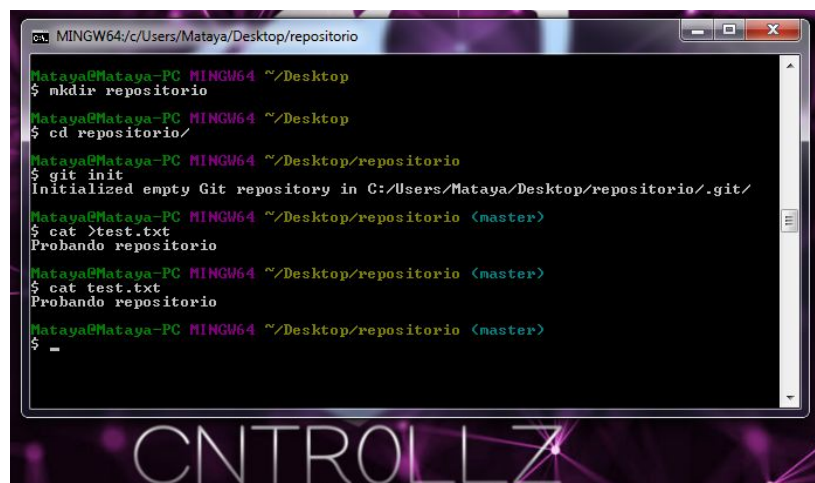
- Windows
 - TYPE test.txt
- Linux
 - cat test.txt
- A continuación mostraremos como se ven los pasos realizados en las distintas terminales, Windows(vea Figura 4), Linux(vea Figura 5),



```
C:\Windows\system32\cmd.exe

C:\Users\Mataya>cd Desktop
C:\Users\Mataya\Desktop>mkdir repositorio
C:\Users\Mataya\Desktop>cd repositorio
C:\Users\Mataya\Desktop\repositorio>git init
Initialized empty Git repository in C:/Users/Mataya/Desktop/repositorio/.git/
C:\Users\Mataya\Desktop\repositorio>Copy Con test.txt
Probando repositorio
^Z
1 archivo(s) copiado(s).
C:\Users\Mataya\Desktop\repositorio>TYPE test.txt
Probando repositorio
C:\Users\Mataya\Desktop\repositorio>_
```

Figura 4. Terminal Windows.



```
Mataya@Mataya-PC MINGW64 ~/Desktop
$ mkdir repositorio
Mataya@Mataya-PC MINGW64 ~/Desktop
$ cd repositorio/
Mataya@Mataya-PC MINGW64 ~/Desktop/repositorio
$ git init
Initialized empty Git repository in C:/Users/Mataya/Desktop/repositorio/.git/
Mataya@Mataya-PC MINGW64 ~/Desktop/repositorio (master)
$ cat >test.txt
Probando repositorio
Mataya@Mataya-PC MINGW64 ~/Desktop/repositorio (master)
$ cat test.txt
Probando repositorio
Mataya@Mataya-PC MINGW64 ~/Desktop/repositorio (master)
$ _
```

Figura 5. Terminal Linux.

Principios básicos.

Hasta aquí solo hemos generado contenido, pero realmente aun no estamos utilizando Git, para ello procederemos a realizar una breve explicación de los fundamentos tras este.

- Añadir archivos en la carpeta de trabajo.
- Comparar el estado actual del repositorio con el anterior.
 - `git status`
- Agregar archivos al repositorio.
 - `git add <nombredelarchivo.extension>`
 - `git add .` (si queremos agregar todos los archivos)
- Identificar version con un comentario, en esta se debemos especificar si estamos resolviendo un error, agregando nuevas características, realizando una prueba entre otros.³
 - `git commit -m "Asunto...\n cuerpo... \n pie\n"`
- Añadimos las credenciales necesarias.
 - `git config --global user.name "cntr0llz"`
 - `$ git config --global user.email cntr0llz@protonmail.com`

Luego de esto, nuestro repositorio cuenta con el registro completo de cada uno de los cambios realizados, vuelva a repetir estos últimos pasos, cada vez que modifique o agregue un archivo.(vea Figura 6)

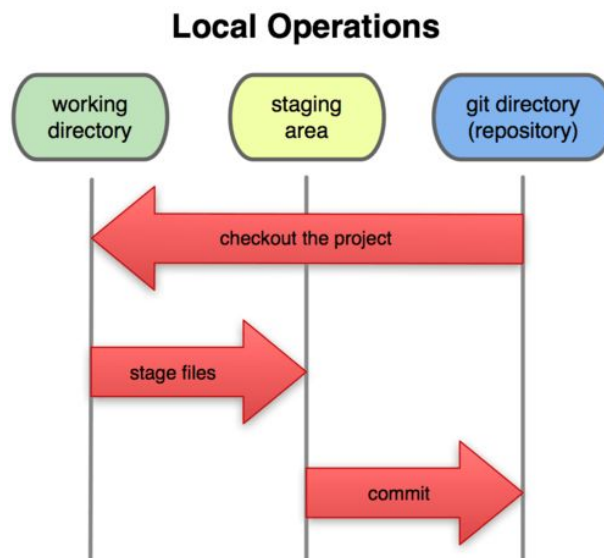


Figura 6. Diagrama Operaciones Locales.

³ Audacity-Git , consultado el (12/05/2019).(<http://udacity.github.io/git-styleguide/>)