

**Evidencia De Conocimiento: GA7-220501096-AA1-EV03 Identifica  
Herramienta De Versionamiento.**

**Jairo Rafael Morales Hernández CC 9290029**

Ficha 2627108

Complejo Tecnológico, Turístico y Agroindustrial Del Occidente Antioqueño –  
Regional Antioquia

Tecnología En Análisis y Desarrollo De Software

Instructor: **Carlos Ernesto Lizarazo Sierra.**

Turbaco – Bolívar

## Introducción

El contenido de este trabajo se basa principalmente sobre las diferencias entre el sistema de control de versionamiento **git local** y **git remoto** y de los principales comandos de esta herramienta, que es fundamental para el desarrollo de mi proyecto software, porque con **GIT**, me permite controlar las versiones de mi proyecto, GIT permite que haya una copia del trabajo del código de cada desarrollador y, al mismo tiempo, es un repositorio local que almacena el historial completo de todos los cambios hechos dentro del código, sin importar que un desarrollador tenga una versión de código más avanzada o atrasada que otro.

## **Objetivos**

1. Saber que es git local y git remoto.
2. Conocer cuáles son las diferencias que existen entre las dos.
3. Aprender a utilizar los comandos de git.
4. Aprender a intercambiar archivos de códigos.
5. Aprender a trabajar en proyectos colaborativos de código abierto.

Comandos básicos de git remoto	
Comandos	Función
\$ git remote	Su función es ver los repositorios remotos que tienes configurados
\$ git remote -v	Muestra las URLs que Git ha asociado al nombre y que serán usadas al leer y escribir en ese remoto.
\$ git remote add [nombre-remoto][url]	Para definir un repositorio remoto y asociarlo a un nombre para su referenciación.
\$ git clone	Se utiliza principalmente para apuntar a un repositorio existente y clonar o copiar dicho repositorio en un nuevo directorio, en otra ubicación.
\$ git fetch [nombre-remoto]	Una vez definido el remote se pueden extraer los datos utilizados con este comando.
\$ git push origin master	Cuando tienes un proyecto que quieres compartir, debes enviarlo a un servidor. El comando para hacerlo es simple: <code>git push [nombre-remoto] [nombre-rama]</code> . Si quieres enviar tu rama master a tu servidor origin (recuerda, clonar un repositorio establece esos nombres automáticamente), entonces puedes ejecutar este comando y se enviarán todos los <b>commits</b> que hayas hecho al servidor:
\$ git remote show [nombre-remoto]	Sirve para ver más información acerca de un remoto en particular.
\$ git remote rename	Si quieres cambiar el nombre de la referencia de un remoto puedes ejecutar este comando. Por ejemplo, si quieres cambiar el nombre de pb a paul, puedes hacerlo así: <code>\$ git remote rename pb paul</code> <code>\$ git remote origin paul</code>
\$ git push [nombre-remoto] [nombre-rama]	Para enviar información desde el repositorio local hacia el servidor remoto.
\$ git remote rm	Si por alguna razón quieres eliminar un remoto – has cambiado de servidor o no quieres seguir utilizando un mirror o quizás un colaborador ha dejado de trabajar en el proyecto – puedes usar este comando. Ejemplo: <code>\$ git remote rm paul</code> <code>\$ git remote origin</code>
\$ git remote add ref https://github.com/paulboone/ticgit	Donde nombre-remoto corresponde al nombre con que se referencia el repositorio y URL es la ubicación lógica del mismo en un entorno de red o en una dirección de internet. En el

	siguiente ejemplo se define un repositorio ubicado en un servidor de Github y cuyo nombre de referencia es ref.
--	---

Comandos básicos de git local	
Comandos	Función
\$ git init	El comando git init crea un nuevo repositorio de Git. Puede utilizarse para convertir un proyecto existente y sin versión en un repositorio de Git, o para inicializar un nuevo repositorio vacío.
\$ git clone https://url_del_repositorio	Este comando creará un nuevo subdirectorio llamado .git, el cual contendrá todos los archivos necesario para el repositorio.
\$ git status	El comando git status muestra el estado del directorio de trabajo y del área del entorno de ensayo. Permite ver los cambios que se han preparado, los que no y los archivos en los que Git no va a realizar el seguimiento.
\$ git add Nombre_archivo	Sirve para rastrear por Git, todos los archivos nuevos.
\$ git add Directorio	Se utiliza para git haga un rastreo de un directotio.
\$ git commit	Un comando de atajo que crea inmediatamente una confirmación con un mensaje de confirmación usado. De manera predeterminada, git commit abrirá el editor de texto configurado localmente y solicitará que se introduzca un mensaje de confirmación.
\$ git commit -m "En esta versión se arreglo e 1 archive w"	Al ejecutar la configuración el sistema desplegará un mecanismo para recibir un mensaje de confirmación que será asociado a esta operación de commit. También es posible agregar el comentario explícitamente en la ejecución de la confirmación usando la opción -m.
\$ git commit -a-m "comentario de esta confirmación"	También es posible ejecutar una operación de confirmación que salte el paso de preparación que se logra con la ejecución del comando add. Es decir, la operación de confirmación se encarga de preparar todos los archivos rastreados y luego confirmar. Esto es posible agregando la opción -a
\$ git log	git log. El comando git log muestra las instantáneas confirmadas. Te permite ver el historial del proyecto, filtrarlo y buscar cambios concretos. Mientras que git status te permite examinar el directorio de trabajo y el entorno de ensayo, git log solo muestra el historial confirmado.

## Características y diferencias de git local y git remoto

	Git local	Git remoto
1	Es un contenedor de archivos que se encuentra en tu ordenador.	Podemos trabajar en cualquier proyecto git.
2	Puedes tener un proyecto lleno de diferentes archivos y es allí donde vas haciendo tus modificaciones y guardándolas para así, ir creando varias versiones.	Los repositorios remotos son versiones de tu proyecto que están hospedados en internet.
3	Cuando creas un repositorio local automáticamente se crea un directorio (.git) que no vas a poder verlo dentro de tus archivos pero que está allí.	Puedes tener varios git y en cada uno de ellos tendrás generalmente permisos de solo lectura o de lectura y escritura.
4	Al inicializar un repositorio en una carpeta estarás diciendo que todos los archivos que están allí dentro formaran parte de un mismo repositorio.	Con git remoto puedes enviar y traer datos cada vez que necesites compartir tu trabajo.
5	Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma o de forma remota con facilidad.	Puedes añadir o eliminar repositorios remotos que ya no son válidos.
6	Git, es un restaurador de contenido.	Git te permite compartir ramas enteras entre repositorios.
7	Git puede ser utilizado para almacenar contenido.	Git remote está diseñado para proporcionar a cada desarrollador un entorno de desarrollo totalmente aislado. Esto significa que la información no se envía automáticamente entre repositorios.
8	Los proyectos de software tienen múltiples desarrolladores trabajando en paralelo. Así que se necesita git para asegurarse de que no hay conflicto de códigos entre ellos.	Git remote admite numerosas formas de hacer referencia a un repositorio remoto. Una de las maneras más fáciles de acceder a uno de ellos es a través de los protocolos HTTP y SSH.