**Resumen de comando [repositorio] [clone] [push] [pull] [branch] [tag] [init] [commit]**

**Kevin Osvaldo Vasquez De Los Santos**

**Instituto Tecnológico de las Américas (ITLA)**

**Tecnólogo en Telecomunicaciones**

**Ramon Antonio Gómez Florián**

**Santo Domingo, República Dominicana**

**24 de ene. del 2025**

**Comando [repositorio] (repository)**

**Definición:**

Un repositorio es un espacio donde se almacena todo el código fuente de un proyecto junto con su historial de cambios. Puede ser local (en tu computadora) o remoto (en una plataforma en la nube como GitHub, GitLab, o Bitbucket).

**Tipos de repositorio:**

* **Local:** Repositorio almacenado en tu sistema local.
* **Remoto:** Repositorio alojado en la nube o servidor remoto para compartir y colaborar.

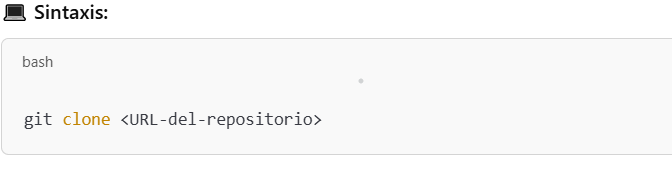
**Funciones principales:**

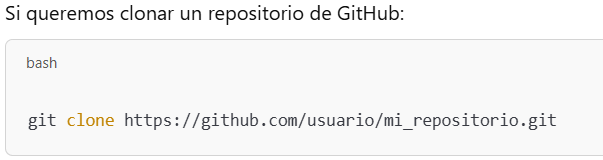
1. **Almacenar archivos:** Permite guardar el código y otros archivos relacionados con el proyecto.
2. **Rastrear cambios:** Mantiene un historial de todas las modificaciones realizadas en el código.
3. **Colaboración:** Facilita el trabajo en equipo sincronizando cambios entre los colaboradores.

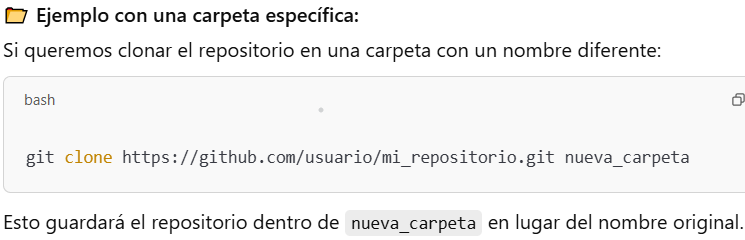
Ejemplo:

**Comando** **[clone]**

**Definición:**  
El comando git clone se utiliza para copiar un repositorio remoto y traerlo a la computadora local. Esto crea una copia exacta del repositorio, incluyendo su historial de versiones, archivos y ramas.

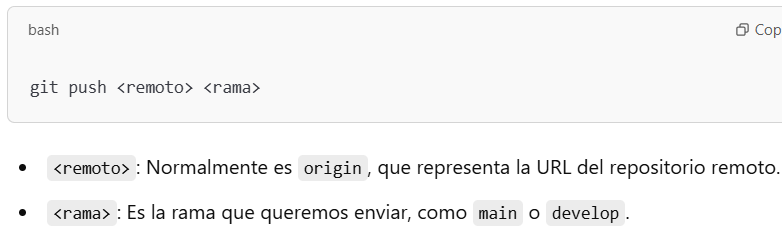
**Ejemplos:**



Esto creará una carpeta llamada mi\_repositorio con todo el contenido del repositorio remoto.

**Comando [push]**

**Definición:**  
El comando git push se usa para enviar los cambios confirmados (commits) desde el repositorio local a un repositorio remoto, como GitHub, GitLab o Bitbucket. Es útil para compartir el trabajo con otros desarrolladores o respaldar el código en la nube.



* **<remoto>:** Normalmente es origin, que representa la URL del repositorio remoto.
* **<rama>:** Es la rama que queremos enviar, como main o develop.

**Ejemplo:**  
Si hemos hecho cambios en la rama main y queremos subirlos al repositorio remoto:

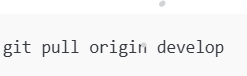
**Ejemplo con una nueva rama:**  
Si hemos creado una nueva rama y queremos subirla al repositorio remoto por primera vez:

La opción -u (o --set-upstream) establece un enlace entre la rama local y la remota.

**Comando [pull]**

**Definición:**  
El comando git pull se utiliza para obtener los cambios más recientes de un repositorio remoto y fusionarlos con la rama actual en el repositorio local. Es una combinación de git fetch (descargar cambios) y git merge (fusionar cambios).

* **<remoto>:** Normalmente es origin, que representa la URL del repositorio remoto.
* **<rama>:** Es la rama de la que queremos traer los cambios, como main o develop.

**Ejemplo:**  
Si queremos actualizar nuestra rama main con los últimos cambios del repositorio remoto:

**Ejemplo con una rama específica:**  
Si estamos trabajando en develop y queremos actualizarla:



**Consejo:**  
Si hay conflictos entre los cambios locales y los remotos, Git pedirá que los resuelvas manualmente antes de completar la fusión.

**Comando [branch]**

**Definición:**  
El comando git branch permite listar, crear y eliminar ramas en un repositorio. Las ramas en Git se utilizan para desarrollar nuevas características o corregir errores sin afectar el código principal.

**Ejemplos:**

1. **Listar todas las ramas disponibles:**

****La rama actual aparecerá resaltada con un \*.

1. **Crear una nueva rama:**

****Esto crea la rama, pero no cambia a ella.

1. **Cambiar a una rama existente:**

O con el nuevo comando recomendado en Git:

1. **Crear y cambiar a una nueva rama en un solo paso:**

O con switch:

1. ** Eliminar una rama local:**

**Si la rama no ha sido fusionada, usa -D para forzar la eliminación:**

****

1. **Ver ramas remotas:**

****

1. **Ver todas las ramas (locales y remotas):**

**Comando [tag]**

**Definición:**  
El comando git tag se usa para crear etiquetas (tags) en el historial de Git. Estas etiquetas marcan versiones importantes del código, como lanzamientos de software (ej. v1.0, v2.0).

**Ejemplos:**

1. **Listar todas las etiquetas existentes:**

****

1. **Crear una nueva etiqueta anotada:**   
   Una etiqueta **anotada** incluye un mensaje, el autor y la fecha.

****

1. **Crear una etiqueta ligera:**

****Las etiquetas ligeras solo son un puntero a un commit, sin información extra.

1. **Ver información de una etiqueta anotada:**
2. **Subir etiquetas al repositorio remoto:**

****

**Para subir todas las etiquetas de una vez:**

****

1. **Eliminar una etiqueta localmente:**

****

1. **Eliminar una etiqueta en el repositorio remoto:**

**Comando [init]**

**Definición:**

el comando git init se utiliza para **inicializar un nuevo repositorio de Git** en un directorio existente. Al ejecutar este comando, Git crea un nuevo repositorio vacío en el directorio actual, lo que te permite empezar a hacer un seguimiento de cambios en tus archivos.

**¿Cómo usar git init?**

1. **Abre tu terminal** o línea de comandos.
2. **Navega al directorio** en el que deseas iniciar el repositorio:
3. **Ejecuta el comando git init** para inicializar el repositorio:

**Otros Ejemplos:**

1. **Añadir archivos al repositorio: Después de inicializar el repositorio, puedes agregar archivos con el comando:** Esto añadirá todos los archivos del directorio al área de preparación de Git.
2. **Hacer tu primer commit**: Luego de añadir archivos, realiza el primer commit: 

### **Resultado**

Al ejecutar git init, ahora tienes un repositorio Git local que puede seguir tus cambios. Desde este punto, puedes continuar con la gestión de tu código con otros comandos de Git, como git add, git commit, y más.

**Comando [commit]**

**Definición:**

El comando git commit en Git se usa para guardar los cambios en el repositorio local. Cada vez que haces un commit, Git registra un "punto de control" con los cambios que has realizado en los archivos del proyecto, junto con un mensaje que describe esos cambios.

Donde:

* **-m** es una opción que se utiliza para agregar un mensaje de commit.
* **"Mensaje del commit"** es una breve descripción de los cambios que has realizado.

**Ejemplos:**

**1. Hacer un commit básico:** Este es el uso más común de git commit. Se agrega un mensaje de commit para describir los cambios que se han realizado.

1. **Hacer un commit con múltiples archivos:** Si modificaste o creaste varios archivos y quieres hacer un solo commit con todos esos cambios, puedes usar git add . para agregar todos los archivos al área de preparación, y luego hacer el commit:



1. **Hacer un commit después de modificar solo un archivo:** Si solo modificaste un archivo específico, puedes hacer un commit solo para ese archivo:



**Conclusión**

En conclusión, los comandos de Git son herramientas esenciales para gestionar proyectos de software de manera eficiente, permitiendo desde la inicialización del repositorio hasta la colaboración con otros desarrolladores. Estos comandos te permiten:

* Gestionar el código local y remoto a través de operaciones como git commit, git push y git pull.
* Organizar tu trabajo en ramas para facilitar el desarrollo paralelo y evitar conflictos.
* Etiquetar versiones importantes del proyecto con git tag, lo que facilita el seguimiento de hitos.

Dominar estos comandos te permitirá trabajar de manera más ágil y estructurada, facilitando la colaboración y el control de versiones en cualquier proyecto de software.