



## Lista de comandos Git

---

A continuación se presenta una lista de comandos importantes para utilizar Git, además de entregar una breve explicación de su uso.

Es necesario recordar que Git es un sistema de control de versiones, es decir, registra los cambios en un archivo o conjunto de archivos a lo largo del tiempo para poder recuperar versiones específicas más adelante. Cada commit es una versión del proyecto y cada versión contiene un ID que lo identifica.

## Control de versiones

---

El control de versiones nos ayuda a:

- Revertir el estado de los archivos a un estado anterior.
- Revertir todo el proyecto a un estado anterior.
- Comparar cambios a lo largo del tiempo.
- Ver quién modificó los archivos y cuándo.
- Evaluar archivos que podrían estar generando un problema.
- Evaluar quién generó un problema y cuándo.
- Recuperar archivos borrados.
- Entre otros

# Estados

---

Existen 3 estados:

1. **Modificado:** Cuando tu proyecto está modificado, o sea, se han editado, borrado o creado archivos dentro del directorio.
2. **Preparado (staged):** Cuando agregas los archivos al área de preparación ( `git add` ).
3. **Commiteado:** Cuando confirmas los cambios. Toma los archivos del área de preparación, y almacena estas modificaciones en tu directorio de git.

# Comandos

---

- `git help` : Muestra los comandos más utilizados de git con una breve descripción.
- `git init` : Inicia git en un directorio, generando un repositorio git. Este comando se ejecuta una vez por proyecto, óptimamente al iniciar el proyecto.
- `git status` : Comprueba el estado del proyecto:
  1. Informa en qué rama se encuentra.
  2. Lista los archivos que se han editado, borrado o creado.
  3. Muestra cuáles archivos están bajo seguimiento (en el área de preparación) y cuáles no.
- `git add` : Agrega archivos al área de preparación.
  - `git add --all` o `git add .` : Agrega todos los archivos modificados al área de preparación.
  - `git add index.html` : Agrega sólo el archivo "index.html" al área de preparación.
  - `git add index.html README.md` : Agrega los archivos "index.html" y "README.md" al área de preparación.
- `git commit -m "mensaje de confirmación"` : Confirma los cambios que están seguidos en el área de preparación y añade un mensaje identificador del commit. Es importante que el mensaje sea descriptivo y significativo para identificarlo más adelante.
- `git log` : Lista los commits realizados en el repositorio, en orden cronológico inverso.
- `git branch` : Lista las ramas que existen en el repositorio.
- `git branch nueva_rama` : Crea una nueva rama.
- `git checkout nombre_rama` : Cambia de rama.
- `git checkout -b nueva_rama` : Crea una nueva rama y automáticamente se cambia a la rama creada.
- `git merge nombre_de_rama` : Trae los cambios de la rama especificada a la rama donde estamos posicionados.
- `git clone url/name.git nombre_proyecto` : Clona un proyecto de git en la carpeta nombre\_proyecto.
- `git push origin nombre_de_rama` : Sube los commits de una rama específica a un repositorio remoto.
- `git pull origin nombre_de_rama` : Hace una actualización en una rama específica local, desde la rama específica del repositorio remoto (útil cuando estamos trabajando con más personas).

Referencia [git-scm.com](https://git-scm.com)