

Sistemas de Control de Versiones



¿Qué es un control de versiones?

- Un sistema de control de versiones (VCS) es una **herramienta que permite registrar los cambios realizados en los archivos** de un proyecto a lo largo del tiempo. Características:
 - Facilita la colaboración entre varios desarrolladores
 - Permite volver a versiones anteriores del código
 - Mantiene un historial detallado de modificaciones.

¿Para qué se usa en el desarrollo web?

En el ámbito del desarrollo web, los sistemas de control de versiones se utilizan para:

- Coordinar el trabajo entre múltiples desarrolladores.
- Mantener un historial de cambios del código fuente.
- Facilitar la implementación de nuevas funciones y la corrección de errores.
- Permitir la integración continua y despliegue automatizado de proyectos.

Tipos de Sistemas de Control de Versiones

- 1. **Sistemas Locales:** Guardan versiones en el mismo equipo. Ejemplo: copias manuales de archivos.
- 2. **Sistemas Centralizados (CVCS):** Un servidor central almacena todas las versiones, y los usuarios trabajan conectados a él. Ejemplo: Subversion (SVN).
- 3. **Sistemas Distribuidos (DVCS):** Cada usuario tiene una copia completa del repositorio. Ejemplo: Git, Mercurial.

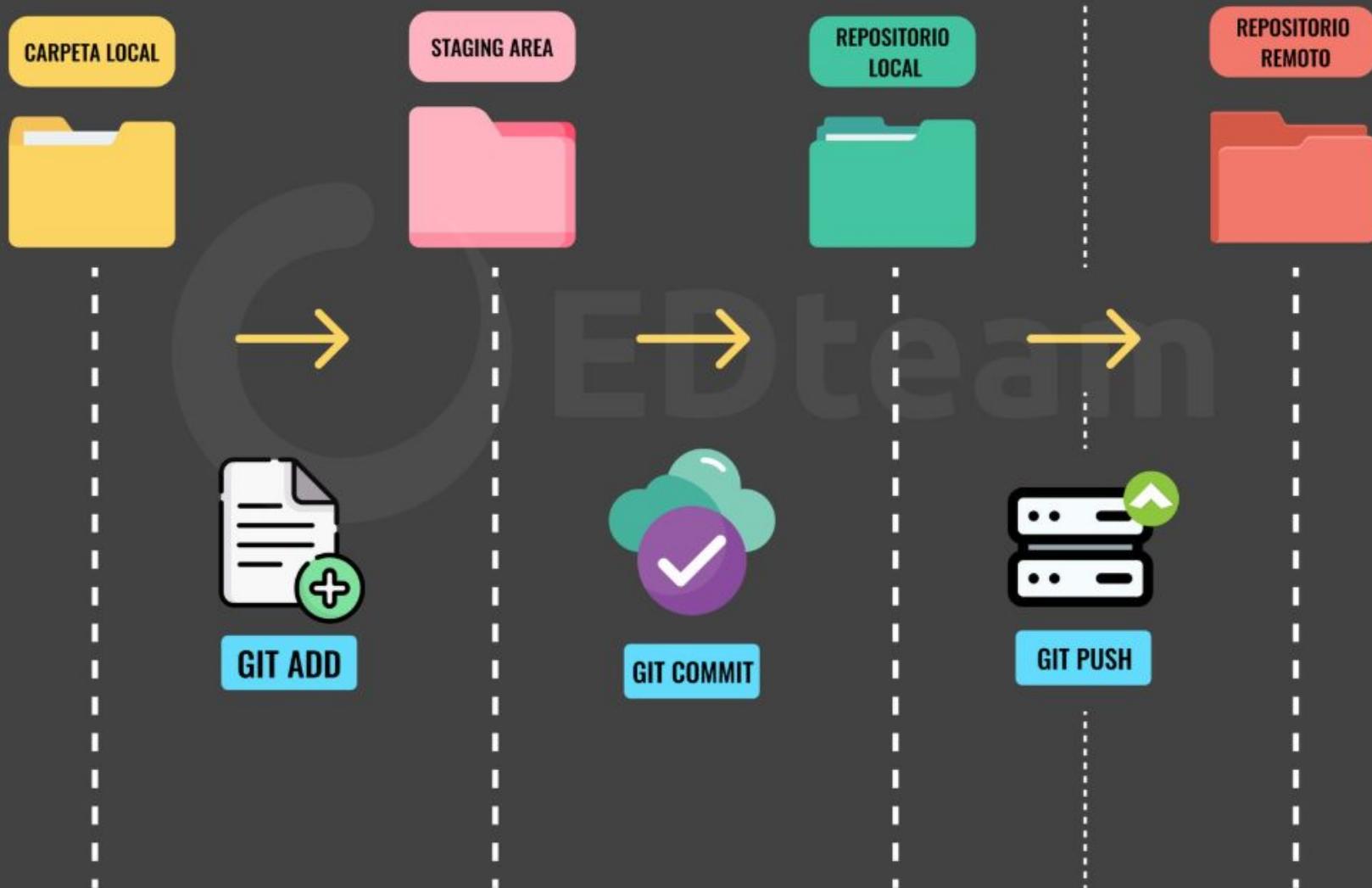
Los tres estados en Git

- **Modificado (Modified)**: El archivo ha sido editado pero los cambios aún no se han guardado en Git.
- **Preparado (Staged)**: El archivo está listo para ser confirmado en el próximo commit.
- **Confirmado (Committed)**: El archivo está guardado en la base de datos de Git.

Secciones de un proyecto Git

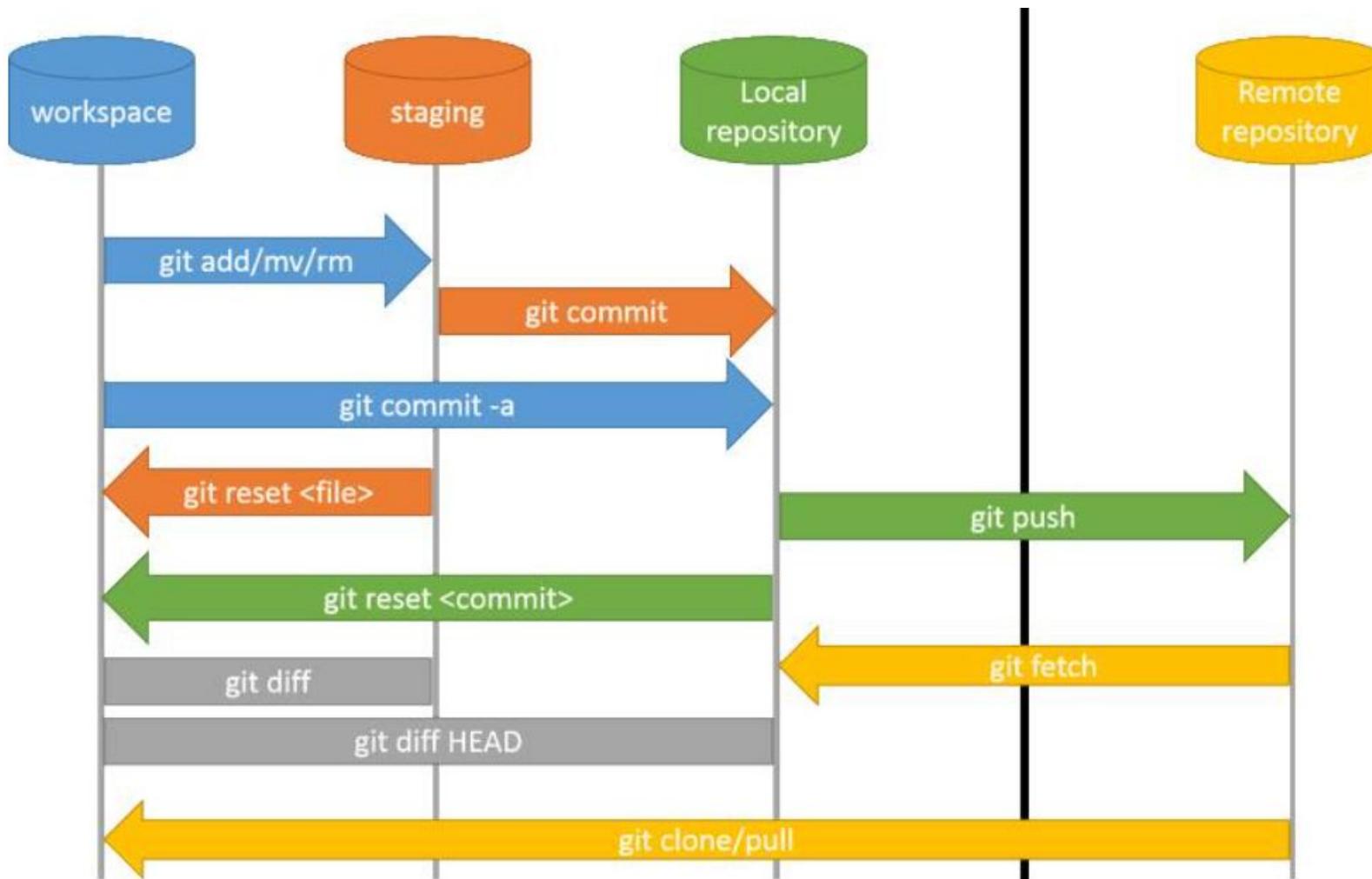
- **Directorio de trabajo (Working Directory):** Donde se editan los archivos reales del proyecto.
- **Área de preparación (Staging Area):** Almacena los cambios que se incluirán en el próximo commit.
- **Repositorio (Repository):** Guarda la base de datos de todos los commits del proyecto.

¿CÓMO FUNCIONA GIT?



Flujo de trabajo en Git

- El flujo de trabajo típico en Git incluye los siguientes pasos:
 - 1. Modificar archivos en el directorio de trabajo.
 - 2. Añadir los archivos modificados al área de preparación con `git add`.
 - 3. Confirmar los cambios con `git commit`.
 - 4. Enviar los commits al repositorio remoto con `git push`.
 - 5. Descargar actualizaciones con `git pull`.



Git y la velocidad

- Se dice que '**los dioses de la velocidad han bendecido Git con poderes sobrenaturales**' porque Git está optimizado para trabajar de forma extremadamente rápida. Al ser un sistema distribuido, la mayoría de las operaciones se realizan localmente sin depender de un servidor, lo que hace que Git sea más eficiente y ágil que otros VCS.