

## Proceso de control de versión con GIT

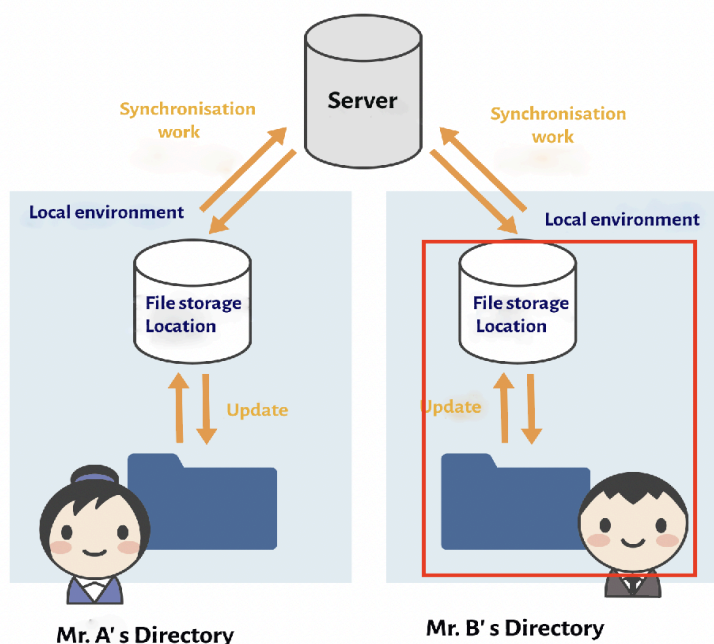


### El control de versiones en Git consta de dos flujos

1. Crear un registro de los cambios localmente (en su propio entorno de PC).
2. Sincronización de los registros de cambios con el remoto (entorno en el servidor al que está conectado cada trabajador).

Este texto describe '1. Creación de registros de cambios locales'.

En concreto, se trata de "añadir cambios a un archivo en el área de preparación y registrarlos como una confirmación en el repositorio local".



## ¿Qué es un repositorio?

Un repositorio es una ubicación de almacenamiento para archivos que están sujetos al control de versiones.

Como Git es un sistema de control de versiones distribuido, replica un repositorio en un servidor (repositorio remoto) como un repositorio en tu PC (repositorio local). Los cambios realizados en el repositorio local no afectan a los demás repositorios hasta que se realiza una operación de sincronización.

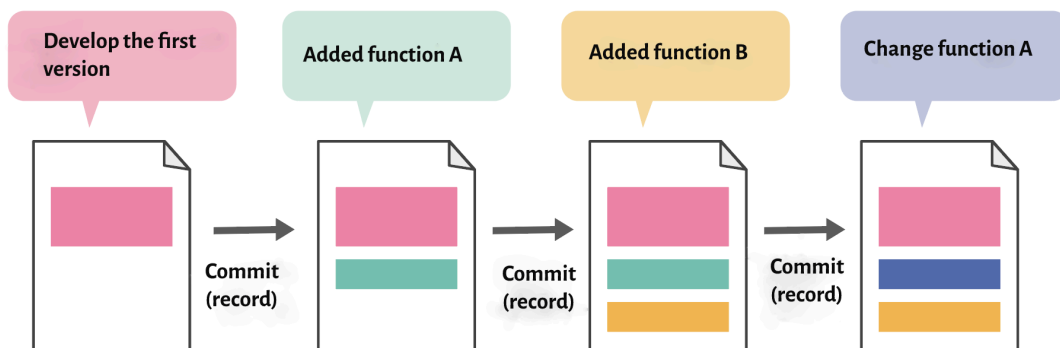
## ¿Que es un commit?

Un commit es un único registro coherente de los cambios realizados en un archivo.

Los commits incluyen la fecha y hora de la actualización, el archivo actualizado y su contenido, y el mensaje en el momento de la actualización.

Esto proporciona visibilidad sobre qué cambios hizo cada trabajador, cuándo y con qué intención.

Además, es frecuente que las características se añadan o cambien como una unidad de commit. Esto se debe a que hace que sea más fácil seguir el historial de commits.



## ¿Qué es una staging area (área de preparación)?

El área de preparación es una zona en la que se almacenan temporalmente los archivos que se van a confirmar. A veces se denomina index (índice).

Para hacer un seguimiento de tu registro de cambios con Git, necesitas añadir el archivo al área de preparación.

Los cambios en el archivo añadido al área de preparación son los cambios en el archivo que contiene la confirmación.

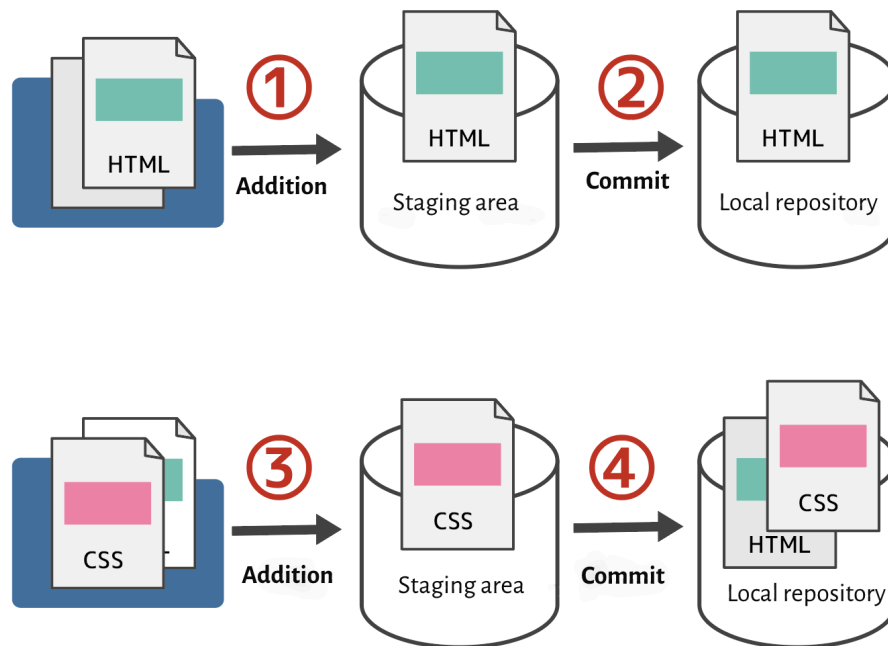
Por lo tanto, ajustar los cambios que añades al área de preparación garantiza que la unidad y la intención de la confirmación sean adecuadamente coherentes.

Esto le permite registrar versiones de cambios que son fáciles de entender en retrospectiva.

Por ejemplo, supongamos que ha actualizado sus archivos HTML y CSS.

En ese caso, para que la gestión de versiones sea más fácil de entender, se utiliza el siguiente proceso.

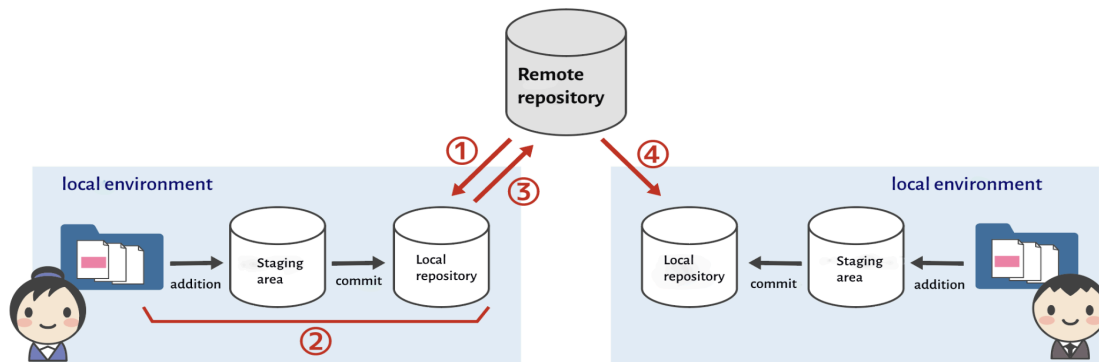
1. Añada el archivo HTML actualizado al área de preparación.
2. Confirmar como una actualización del archivo HTML.
3. Añada el archivo CSS actualizado al área de preparación.
4. Confirmar como una actualización del archivo CSS.



## ¿Cómo utilizar los repositorios?

Basándose en lo anterior, en los lugares de trabajo de desarrollo reales se utiliza el siguiente control de versiones.

1. Replicar el contenido de un repositorio remoto en el entorno del trabajador como repositorio local.
2. Los trabajadores añaden cambios de archivos locales al área de preparación en unidades apropiadas y los registran como commits
3. Los commits registrados se reflejan en el repositorio remoto mediante operaciones de sincronización.
4. Otros trabajadores también pueden sincronizarse para reflejar los cambios en sus repositorios locales.



## Resumen

En el control de versiones con Git, el flujo de trabajo es "1. crear registros de cambios localmente" y "2. sincronizar los registros de cambios con el remoto".

El registro local de cambios se realiza añadiendo commits al repositorio, que registran los cambios en los archivos en un único segmento.

Los archivos que deben confirmarse son los añadidos al área de preparación.

Añada los archivos modificados al área de preparación en unidades apropiadas o regístrelos como una confirmación con un mensaje. Esto permite registrar la versión modificada de forma que sea fácil de comprender.



Documentación oficial de Git [conceptos básicos](#)