

UTILIDAD DEL SISTEMA DE CONTROL DEL VERSIONES – GITHUB

“El sistema de control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante”. (Chacon & Straub, 2018, p. 9)

Partiendo de esta definición y en el contexto de la ingeniería de sistemas, esta herramienta de GITHUB permite tener un repositorio compartido donde varios usuarios interactúan en el desarrollo de un proyecto, mantener actualizado y sincronizado los avances de los usuarios sobre el proyecto, llevar un control de los diferentes cambios y versiones, hacer copias de seguridad y restaurarlas, entre otras utilidades.

Git facilita el control de versiones a nivel local y GitHub en línea, para el control local esta herramienta tiene una base de datos que almacena todos los datos y registros relacionados con los cambios y modificaciones realizados al interior del repositorio local, manejando una línea de tiempo de todos los eventos relacionados con el Repositorio. Esta información solo se verá a nivel local si no se sincroniza con la plataforma web.

Chacon & Straub (2018) afirma que:

Git tiene tres estados principales en los que se pueden encontrar tus archivos: confirmado (**committed**), modificado (**modified**), y preparado (**staged**). Confirmado significa que los datos están almacenados de manera segura en tu base de datos local. Modificado significa que has modificado el archivo pero todavía no lo has confirmado a tu base de datos. Preparado significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación. (p. 16)

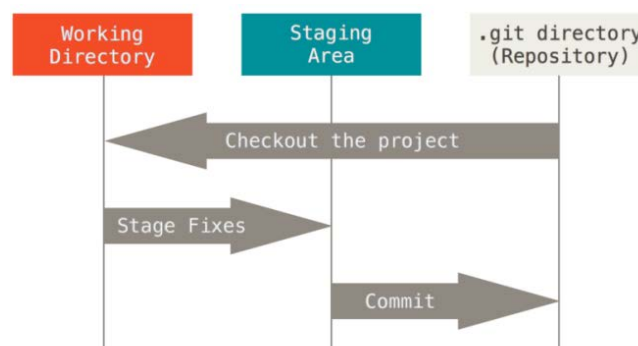


Ilustración 1. Directorio de trabajo, área de preparación y directorio de Git.

Fuente: (Chacon & Straub, 2018, p. 16)

Para finalizar este texto, informamos que proyectamos en otra ocasión, hablar de los estados de secciones principales de un proyecto de Git: el directorio de Git (Git directory), el directorio de trabajo (working directory), y el área de preparación (staging area).

Referencia

Chacon, S. & Straub, B. (2018). *Pro GIT*. Mountain View: Apress.