

Sintaxis y Semántica de los Lenguajes

Conceptos elementales de Git / GitHub

A continuación les comparto un breve resumen para ir comprendiendo cuestiones referidas a git / gitHub. Comencemos !

¿Qué es Git?

Git es un sistema de control de versiones (VCS) popular, gratuito y de código abierto diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

¿Qué es GitHub?

GitHub es un sistema de control de versiones basado en la web que ofrece un servicio de hosting de repositorios almacenados en la nube bajo la misma lógica de Git. Esencialmente, hace que sea más fácil para individuos y equipos usar Git como la versión de control y colaboración dado que cualquier persona puede inscribirse y hospedar un repositorio de código público de manera completamente gratuita, lo cual hace que GitHub sea especialmente popular con proyectos de fuente abierta.

¿Qué es un repositorio?

El repositorio es una estructura de datos utilizada por VCS (Sistema de control de versiones) para almacenar metadatos de un conjunto de archivos y directorios referidos a sus modificaciones llevando un historial de todos los cambios realizados a lo largo del tiempo.

Componentes básicos de Git

Para comenzar a entender su funcionamiento, debemos comprender que tendremos diferentes áreas: Working directory (Directorio de trabajo), Staging área (Área de preparación) y el repositorio (Repository). A continuación detallamos cada una de ellas,

❖ *Working directory*

El directorio de trabajo es simplemente el directorio local actual en el que está trabajando en su computadora local. Es la carpeta donde se encuentran los archivos, independientemente de todo registro que pueda estar llevando Git para el control de versiones. Debemos saber que dentro de ese directorio de trabajo hay una carpeta oculta con el nombre “.git” que almacena toda la información referida al control de versiones que será utilizada para confirmar los cambios en el repositorio posteriormente.

❖ **Staging area**

El staging area (área de preparación) se describe como una vista previa del próximo commit (confirmación). Es decir, cuando utilizemos el comando `git commit`, git tomará los cambios que están en la staging area y hará el nuevo commit de esos cambios. Un uso práctico del staging area es que permite ajustar las confirmaciones permitiendo selectividad respecto de aquellos cambios que queremos confirmar y aquellos que aún queremos conservar en nuestra área de trabajo (working area).

En pocas palabras, la staging area es un espacio intermedio entre nuestra área de trabajo y el repositorio que nos permite controlar aquellos cambios que deseamos confirmar en nuestro repositorio. Más adelante iremos comprendiendo su practicidad e importancia.

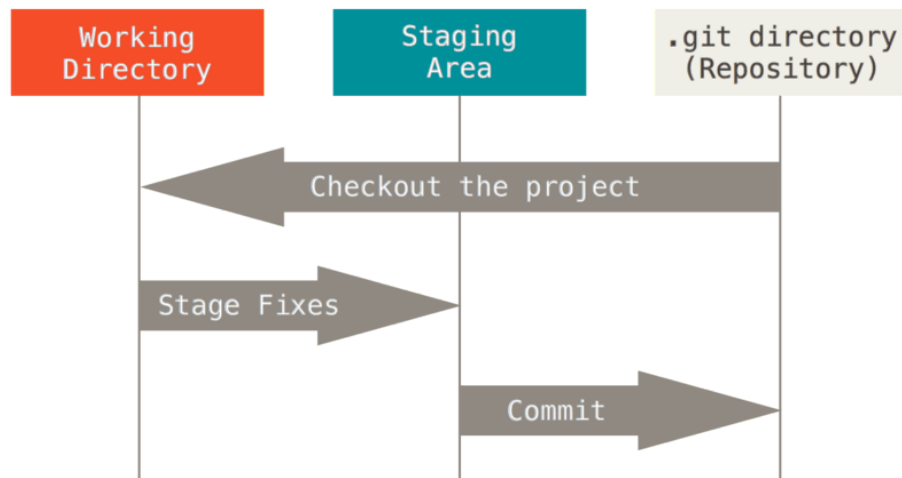
❖ **Repository**

El repositorio es una estructura de datos utilizada por VCS (Sistema de control de versiones) para almacenar metadatos de un conjunto de archivos y directorios referidos a sus modificaciones llevando un historial de todos los cambios realizados a lo largo del tiempo. Un repositorio es como una carpeta para tu proyecto. El repositorio de tu proyecto contiene todos los archivos de tu repositorio y almacena el historial de revisión de cada archivo. Git utiliza un índice para poder realizar el control de versiones Git Index que detallamos a continuación.

❖ **Git Index**

Git index se usa como un área de ensayo (staging area) entre el directorio de trabajo (working directory) y el repositorio (repository). Se utiliza dicho índice (index) para crear un conjunto de cambios que se desean confirmar (commit) juntos en el repositorio. Cuando crea un commit, lo que se confirma es lo que está actualmente en el índice, no lo que está en el directorio de trabajo.

Esquema de trabajo desde Working Directory hasta Repository (local)



La imagen muestra que todo lo que trabajemos de manera local en nuestro proyecto pasará siempre por staging area antes de impactar en el repositorio. Como se puede apreciar, todo cambio en el working directory pasará, en principio, por la staging area.

Flujo de trabajo en Git

El flujo de trabajo básico en Git es algo así:

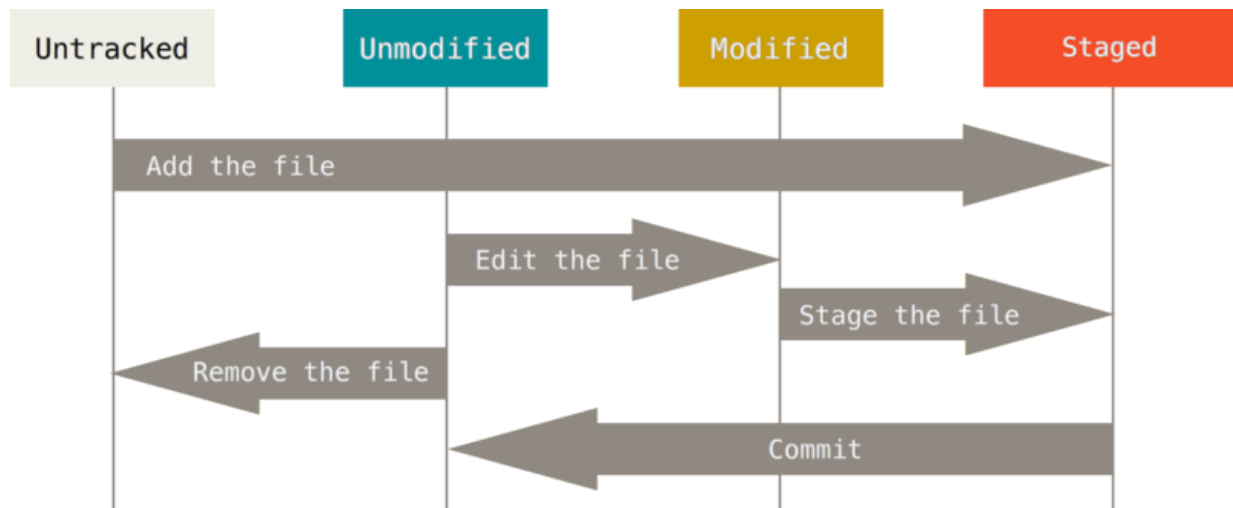
1. Modificamos una serie de archivos en tu directorio de trabajo (working directory).
2. Preparamos los archivos, añadiéndolos a tu área de preparación (staging area).
3. Confirmamos los cambios (commit), lo que toma los archivos tal y como están en el área de preparación y almacena esa copia instantánea de manera permanente en tu directorio de Git.

Posibles estados para los archivo

Los estados posibles para cada archivo son los siguientes

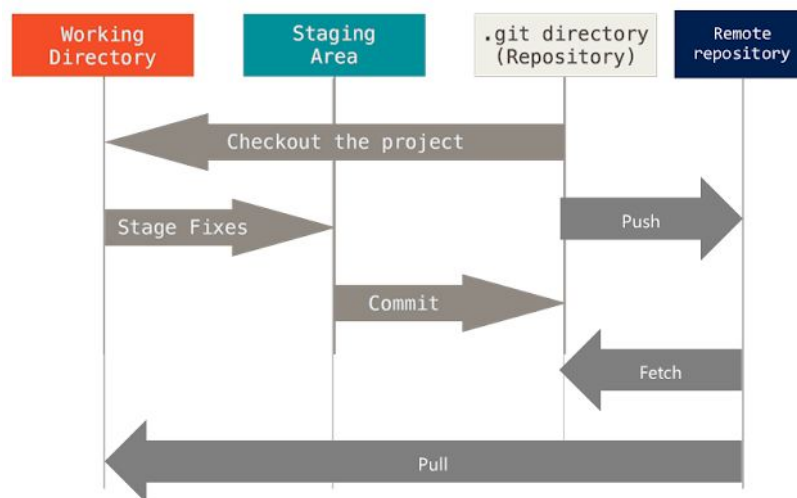
- Si una versión concreta de un archivo está en el directorio de Git, se considera confirmada (**committed**).
- Si ha sufrido cambios desde que se obtuvo del repositorio, pero ha sido añadida al área de preparación, está preparada (**staged**).
- Y si ha sufrido cambios desde que se obtuvo del repositorio, pero no se ha preparado, está modificada (**modified**).
- Si es un archivo que se ha agregado al directorio de trabajo sin haber sido obtenido del repositorio y aún no se ha preparado (**untracked**).

Esquema de trabajo entre los archivos que se encuentran en la carpeta asociada a git que aún no han sido agregados (untracked) o aquellos que ya estando asociados a git han sufrido modificaciones.



En ambos casos utilizaremos el comando `git add` para reflejar en el stage área las modificaciones que realizamos en la working area (área de trabajo)

Esquema de trabajo general e Working Directory hasta Remote Repository



Esta imagen muestra el proceso completo desde lo que cada uno trabaja en la carpeta asociada a su repositorio local hasta su carga en el repositorio remoto (gitHub) y viceversa. En este esquema ya indica cómo se trabajar con un repositorio compartido en la nube.