**Fundamentos de Git.**

Entonces, ¿qué es Git en pocas palabras? Es muy importante asimilar esta sección, porque si entiendes lo que es Git y los fundamentos de cómo funciona, probablemente te sea mucho más fácil usar Git de manera eficaz. A medida que aprendas Git, intenta olvidar todo lo que puedas saber sobre otros VCSs, como Subversión y Perforce; hacerlo te ayudará a evitar confusiones sutiles a la hora de utilizar la herramienta. Git almacena y modela la información de forma muy diferente a esos otros sistemas, a pesar de que su interfaz sea bastante similar; comprender esas diferencias evitará que te confundas a la hora de usarlo.

**Introducción.**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Un sistema de control de versiones trata de un sistema que registra todos los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que podemos recuperar cualquier versión específica más adelante. El control de versiones se puede realizar tanto para versiones de código fuente como para casi cualquier tipo de archivo.

Tenemos diferentes sistemas de control de versiones:

* Sistemas de control de versiones locales.
* Sistemas de control de versiones centralizados.
* Sistemas de control de versiones distribuidos.

Como GIT se trata de un sistema de control de versiones distribuido, vamos a tratar un poco de que trata este sistema de control de versiones.

En un DVCS (Distributed Versión Control Systems) cuando el cliente descarga archivos del repositorio, no sólo descarga la última instantánea de los archivos, también replica completamente el repositorio. De esta manera, si un servidor muere y hay clientes que estaban colaborando con él, cualquiera de los repositorios de cualquier cliente puede copiarse en el servidor para restaurarlo.

Cada vez que se hace una descarga de una instantánea, estaríamos haciendo una copia de seguridad completa de todos los datos del repositorio.

**La mayoría de las operaciones son locales.**

GIT sólo necesita archivos y recursos locales para operar. Como tenemos toda la historia del proyecto en nuestro disco local, la mayoría de las operaciones se realizan de forma muy rápida en comparación con los demás sistemas VCSs.

Para obtener la historia de un proyecto, sólo tendríamos que leer de nuestra base de datos local.

- Para ver cambios introducidos en la versión actual de un archivo y la versión de este mismo archivo hace un mes, GIT puede hacer el cálculo de diferencias localmente.

- Si no tenemos internet, podremos trabajar de forma normal hasta obtener conexión de internet y subir los nuevos datos.

**Integridad.**

Todo en Git es verificado mediante una suma de comprobación antes de ser almacenado, de esta forma es imposible cambiar contenidos de cualquier archivo o directorio sin que GIT lo sepa.

**Estados.**

Los archivos en GIT se pueden encontrar en tres estados diferentes:

* Confirmado: Los datos están almacenados de forma segura en tu base de datos local.
* Modificado: El archivo ha sido modificado, pero no se ha confirmado a tu base de datos local.
* Preparado: Se trata de marcar un archivo modificado como preparado para que vaya a la base de datos local en tu próxima confirmación.

En un proyecto de GIT nos encontramos con tres áreas diferentes:

* El directorio de trabajo: Copia de una versión del proyecto.
* El área de preparación: Se trata de un archivo que almacena la información acerca de lo que va a ir en tu próxima confirmación.
* El directorio de GIT: donde GIT almacena los metadatos y la base de datos de objetos para tu proyecto.

**Instantáneas, no diferencias.**

La principal diferencia entre Git y cualquier otro VCS (Subversión y compañía incluidos) es cómo Git modela sus datos. Conceptualmente, la mayoría de los demás sistemas almacenan la información como una lista de cambios en los archivos. Estos sistemas (CVS, Subversión, Perforce, Bazaar, etc.) modelan la información que almacenan como un conjunto de archivos y las modificaciones hechas sobre cada uno de ellos a lo largo del tiempo.

**Git en un Servidor.**

Ahora vamos a proceder a alojar nuestro proyecto realizado localmente en GitHub. GitHub es un servicio de alojamiento de repositorios de software que ofrece opciones muy interesantes para trabajar en equipo. GitHub está basado en Git.

La ventaja de GitHub es la facilidad de manejo de proyectos, colaboradores y por supuesto la posibilidad que ofrece para contribuir con aquellos que encontremos más interesante. Sin duda se ha convertido en un punto de encuentro para los desarrolladores.

Documento presentado por: *Jessica Amanda Bonilla Rodríguez*