

Git

Git es un sistema distribuido de control de versiones que es altamente utilizado en el desarrollo de software. Fue creado por Linus Torvalds en el año 2005 y se ha convertido en el estándar industrial para el control de versiones. Git permite a los desarrolladores mantener un registro de los cambios realizados a su código a través del tiempo, colaborar con otras personas e incluso revertir hacia una versión previa de este si fuese necesario. En este ensayo se hablará acerca del control de versiones, estados de un archivo, comandos básicos y cómo crear un repositorio utilizando GitHub.

Control de versiones

El control de versiones es el proceso de manejar los diferentes cambios que se realizan a un archivo o conjunto de archivos a través del tiempo. Es esencial en el desarrollo de software, pues permite a los desarrolladores de software trabajar colaborativamente en el mismo código, llevar un control de los cambios realizados (Adiciones, modificaciones y eliminaciones de líneas de código) y llevar un historial de estos cambios, acompañados de cuándo y quién los realizó. Git es un sistema distribuido de control de versiones, que significa que cada usuario puede tener una copia completa del repositorio del código, incluyendo el historial entero de cambios. Git permite a los desarrolladores trabajar de forma offline y unir sus cambios con los cambios de otros desarrolladores cuando estos decidan o bien, cuando regresen a estar en línea.

Estados de un archivo

En Git, un archivo puede existir en tres lugares:

- En el directorio de trabajo
- Área de Staging
- Repositorio

El directorio de trabajo es en donde los desarrolladores hacen cambios al archivo. El área de staging es en donde los cambios están listos y marcados para realizar un commit y el repositorio es en donde los cambios a los que se les hizo un commit se almacenan.

Dicho lo anterior, existen tres estados que pueden poseer un archivo:

- Modificado (Modified)
- Preparado (Staged)
- Confirmado (Committed)

El flujo de trabajo es el siguiente:

1. Se modifica una serie de archivos en el directorio de trabajo.
2. Se preparan los archivos, añadiéndolos al área de preparación o Staging area.
3. Se confirman los cambios, lo que toma los archivos tal y como están en el área de preparación o Staging area y almacena esa copia instantánea de manera permanente en el repositorio.

Comandos básicos de Git

Git tiene un gran número de comandos básicos que los desarrolladores utilizan para manipular el repositorio. Algunos de los más comunes incluyen:

- **git init:** Este comando inicializa un nuevo repositorio de Git en el directorio actual
- **git add:** Este comando prepara los cambios del o los archivos modificados, marcándolos para que se incluyan en el siguiente commit. Se puede utilizar directamente el nombre del archivo modificado que se desea marcar o "." para incluir todas las modificaciones realizadas.
- **git commit:** Este comando hace un "commit" o confirma los cambios en el historial del repositorio.
- **git push:** Este comando envía los cambios confirmados a un repositorio remoto, normalmente en alguna forja como Github.
- **git pull:** Este comando extrae todos los cambios realizados de un repositorio remoto y los aplica en el repositorio local.
- **git clone:** Este comando crea una copia local de un repositorio remoto.

Como crear un repositorio en GitHub

Crear un nuevo repositorio en Github es un proceso sencillo, estos son los pasos:

1. Iniciar sesión en Github, si no se tiene una cuenta, crear una en el apartado de Registro.
2. Al estar en el Dashboard, dar click en "Nuevo repositorio".
3. Ingresar el nombre del nuevo repositorio y una descripción que puede ser opcional.
4. Seleccionar si el repositorio tendrá visibilidad pública o privada.
5. Si se desea crear el repositorio con un archivo con contenido inicial, se puede dar click en "Iniciar este repositorio con un README"
6. Dar click al botón de "Crear repositorio"

Una vez que el repositorio se creó, se puede utilizar el comando **git clone** para crear una copia local en la computadora, realizar cambios, agregarlos al área de stage con **git add**, confirmarlos o hacerles "commit" con **git commit** y enviarlos de vuelta a Github con **git push**.

En conclusión, Git es una herramienta muy útil para el control de versiones, permitiendo a los desarrolladores administrar los cambios realizados a su código y poder colaborar con otros desarrolladores sobre un mismo proyecto. Entendiendo los conceptos y comandos básicos de Git es esencial para cualquier desarrollador, y crear un repositorio en Github es una excelente manera de empezar con Git