

¿Qué es GIT?

Es un sistema de control de versiones, es un sistema o programa que administra las distintas versiones de un programa. Cuando un programador usa git, es porque quiere controlar un proyecto que va a crecer a medida que el tiempo transcurre, y va a hacer cambios o agregando archivos, mejorando el código y demás. Y para controlar esos cambios se usa estos sistemas de control de versiones. Te ayuda al tener un listado con los cambios realizados o colaborar con desarrolladores por todo el mundo. Es un sistema de control distribuido, ósea todos los que trabajen en un código, van a tener una copia local del código, siempre vamos a tener los últimos cambios que estén afectando el código principal. Te ayuda a manejar los cambios cuando haces una alteración en el código, y guarda todas las alteraciones hechas en un código por si se quiere volver a versiones anteriores. Se puede ver que día se modificó el código, quien lo modificó, a qué hora o que línea de código, todo eso se maneja con git. Se pueden revertir los cambios. Puede trabajar con repositorios locales y remotos, ósea un computador que administra tu código, lo vamos a guardar en un computador y va a ser accesible a los desarrolladores que nosotros queramos.

Como usar GIT

Te probé una manera de seguir los cambios de tu código, usando snapchoes. Toma fotos de tu código y luego puedes verlo o alterarlo, copiarlo, modificarlo. Es útil usarlo cuando estamos en la versión 1 de nuestra aplicación.

Comandos básicos de GIT

Es una herramienta que interactúa directamente con el sistema operativo. Por lo tanto lo vamos a usar mucho mediante la consola. GIT te instala un terminal, una consola con mejores funcionalidades. En GIT hay tres estados.

1 WORKING DIRECTORY: Es donde vamos a estar trabajando con nuestros archivos.

2 STAGING AREA: Donde vamos a agregar los archivos que preparemos para el guardado.

3 REPOSITORY

COMANDOS BASICOS:

git init: Es para decir que vamos a crear un proyecto nuevo, o importar un código donde no usemos git.

git add: Es para pasar los archivos del working directory al staging area.

git status: Es para ver en qué estado están nuestros archivos. Si están en el working directory, en el staging area, etc.

git commit: Es para pasarlo del staging area al repository. Para crear una primera versión del código.

git push: Es para subirlo a un repositorio remoto. Ósea para subirlo a un servidor para que otros desarrolladores con nuestro permiso puedan modificar el código.

git pull: Es para en caso que trabajemos con múltiples desarrolladores te trae los cambios que hacen los otros desarrolladores.

git clone: Es para hacer una copia desde el servidor central donde está el código a nuestro computador para empezar a trabajar. Y pegar el url del código desde github.

git checkout – (nombre archivo) : vuelven a la anterior versión del código.

git diff (nombre archivo): para ver las diferencias hechas en el archivo.

git log: Nos muestra los datos de los cambios realizados.

Head =>master: Nos dice en que versión del código estamos trabajando.

(Archivo) .gitignore: Dentro de este archivo escribimos las carpetas que queremos que GIT ignore.

git commit (espacio)–m(espacio) “escribimos mensaje”: Así nos ahorramos que entre en la consola y escribir el mensaje ahí.

git branch: nos muestra las carpetas o proyectos.

git branch (nombre de la versión alternativa): creamos una nueva versión alternativa con el nombre que le demos.

git checkout (nombre de la versión alternativa a la que queramos movernos): no movemos a la otra versión y nos aparece en verde la versión en la que estamos.

git add(espacio)●(punto): se usa para agregar todos los archivos que tengo dentro del entorno de trabajo, sin necesidad de poner los nombres.

Las carpetas vacías GIT las ignora, por eso no aparecen cuando ponemos git status.