

Git Hub

Introducción a GitHub.

- ¿Qué es Git hub?

Es una plataforma de desarrollo en la cual se pueden alojar proyectos en la nube utilizando un sistema de controlador de versiones conocido como Git. Este sistema se encarga de almacenar y administrar el código que a su vez va creando un registro de los cambios que se van agregando al proyecto.

- ¿Qué es Git?

Es un software controlador de versiones con el propósito de registrar los cambios que se le realizan a un código, así como llevar a cabo una coordinación del flujo del trabajo. Este controlador a su vez permite a los desarrolladores descargar una copia del código a sus repositorios locales.

- Comandos básicos de Git

Git clone → Descarga el código fuente desde un repositorio remoto de la última versión y la almacena en el ordenador.

Git Branch → Se utiliza una rama con propósito de que varios desarrolladores trabajen en el mismo proyecto de manera simultánea.

Git checkout → Se utiliza para moverse de una rama a otra dentro de nuestro proyecto en Git.

Git status → Se solicita toda la información de la rama para conocer si está actualizada, hay algo por confirmar, enviar o recibir, si hay archivos en preparación o si hay modificaciones y eliminaciones.

Git add → Se utiliza para incluir todos los cambios realizados en la siguiente petición (commit).

Git commit → Se establece un punto de control dentro de nuestro código y se suben los cambios a la nube al cual se puede volver en caso de ser necesario

Git push → Una vez que se confirmaron los cambios que se realizaron se deben de enviar los cambios al servidor remoto para que almacene los nuevos cambios.

Git pull → Utilizado para recibir las ultimas actualizaciones de nuestros proyectos así se toman todas las actualizaciones del repositorio remoto en nuestro repositorio local

- Repositorios

Es un lugar de almacenamiento en el cual se puede alojar código, archivos, historial y versiones de cada uno de ellos de manera remota donde los desarrolladores tienen un acceso de forma sencilla.

Se puede trabajar con una rama única, pero existe la posibilidad e trabajar con múltiples ramas lo que permitirá que los desarrolladores puedan trabajar en el mismo proyecto de manera simultanea sin afectar el trabajo de sus colegas. Sin embargo, el hecho de crear estas ramas no quiere decir que cada uno tenga que hacer un proyecto diferente totalmente ya que git cuenta con un comando denominado como “Git merge” el cual se utiliza para fusionar una o mas ramas que se tengan dentro de la rama principal.

Por otra parte, en los repositorios existen también los conflictos ya que el trabajo de cada desarrollador dentro de las diferentes ramas puede generar confirmaciones de cambios que se contraponen por lo general la plataforma de Git resuelve estas diferencias y las fusiona de manera automática, sin embargo no siempre es así ya que si ponemos como ejemplo que dos desarrolladores realicen cambios dentro de la misma línea de código por lo que se deben resolver estos conflictos antes de subir los cambios al repositorio remoto para evitar la posible pérdida de información debido a estas situaciones.

- Comandos Avanzados.

Git pr → El comando de pull request se refiere a la acción en la que un desarrollador solicita que se realice una revisión de los cambios que realizo antes de hacer un commit en la rama que se encuentra o en la rama principal.

Bifurcación (Fork) → Una bifurcación se refiere a la creación de un repositorio nuevo que comparte la última versión de un original en el cual se suelen proponer y probar nuevas ideas o cambios antes de anexarlos en el proyecto original

Git rebase → Este permite realizar cambios en los commit del historial dentro del repositorio que se está utilizando.

Git stash → Se utiliza comúnmente para guardar los cambios sin necesidad de realizar un commit dentro del código, por lo regular se utiliza cuando se tiene que pasar de una rama a otra, este guarda los cambios de manera temporal dentro de tu repositorio local.

Git clean → Este comando suele ser muy útil ya que nos ayuda a eliminar los archivos que no tienen seguimiento que se encuentren en el directorio o archivos no deseados.

Git Cherry-pick → Es un comando que es muy sensible y se tiene que ser cuidadoso a la hora de aplicarlo ya que este toma uno o más commits que se hayan realizado y los añade en el programa main actual, usualmente se utiliza para la rápida corrección de errores.