GIT Y GITHUB

¿QUE ES GIT?  
  
Git, un sistema de control de versiones distribuido el cual facilita el registro y la gestión de cambios en archivos, sobre todo de código, a lo largo del tiempo.  
  
Se usa para esto:  
  
Guardar un historial de los cambios que se han hecho.  
  
Permitir el trabajo conjunto sin estorbar el trabajo de otros.  
  
Favorecer un desarrollo en equipo, además, más seguro.

¿QUE ES GITHUB Y QUE LO DIFERENCIA DE GIT?

GitHub es la plataforma en la nube que aloja repositorios Git y permite colaborar, compartir código, hacer revisiones y gestionar proyectos. Mientras que Git era la herramienta de control de versiones que funciona en el computador

Git es la herramienta, GitHub es el servicio que la complementa.

VENTAJAS DE USAR CONTROL DE VERSIONES EN UN PROYECTO

* Mantiene un registro completo de cambios
* Recuperar versiones anteriores si hay errores.
* Facilita trabajar en equipo con tareas específicas para cada uno y a su vez, facilita trabajar de forma remota
* Fortalece la estructura y protege el proyecto

COMANDOS EXPLICADOS  
git init: Inicializa un nuevo repositorio Git en la carpeta actual.

git add: Agrega cambios al área de staging (preparar antes de confirmar).

git commit: Guarda los cambios en el historial del repositorio con un mensaje.

git status: Muestra el estado del repositorio (archivos modificados, pendientes de agregar, etc.).

git log: Muestra el historial de commits.

git push: Envía los cambios del repositorio local al remoto.

git pull: Descarga cambios del repositorio remoto y los integra en el local.

git clone: Crea una copia local de un repositorio remoto.

¿Qué es un repositorio remoto y como se conecta con el local?

Un repositorio remoto es simplemente una versión de tu proyecto guardada en un servidor en internet (o en una red local), a la que varias personas pueden acceder.

Buenas prácticas al trabajar con Git

Hacer commits pequeños y frecuentes con mensajes claros.

Usar ramas (branches) para nuevas funcionalidades o pruebas.

Mantener la rama principal (main o master) estable.

Escribir mensajes de commit descriptivos.

Hacer pull antes de trabajar para tener la última versión.

Ignorar archivos innecesarios con .gitignore.

Revisar git status seguido para evitar confusiones.