***TEORÍA GIT:***

Git es un sistema de control de versiones (SCVD, Sistema de control de versiones distribuida), creado por linus torvalds, su principal objetivo es gestionar las diferentes versiones de aplicaciones.

Un SCVD como Git registra todos los cambios realizados en los archivos a lo largo del tiempo. Esto te permite:

* Llevar un historial completo del desarrollo del proyecto.
* Comparar diferentes versiones para ver qué se modificó.
* Saber quién realizó cada cambio.
* Revertir el proyecto entero a un estado anterior si es necesario.

En esencia, Git te da la capacidad de rastrear, gestionar y colaborar en proyectos de software de forma organizada y segura.

***CONCEPTOS BÁSICOS DE GIT***

**Repositorio:** Corazón de Git. Guarda todo el proyecto y su historia. Puede ser local o remoto (en línea).

**Commit:** Instantánea del proyecto en un momento dado. Guarda el estado actual.

**Log:** Historial de todos los commits, mostrando quién, qué y cuándo se modificó.

**Branch (Rama):** Versión paralela del proyecto para trabajar en cambios sin afectar la línea principal.

**Staging area (Área de preparación):** Zona intermedia para seleccionar cambios antes de guardarlos con un commit.

**Merge (Fusión):** Unir los cambios de una rama a otra. A veces requiere resolver conflictos.

**Cherry-pick:** Tomar commits específicos de una rama y aplicarlos a otra.

**.gitignore:** Archivo para decirle a Git qué archivos ignorar (temporales, etc.).

**Pull:** Traer los últimos cambios del repositorio remoto y unirlos a tu rama local.

**Push:** Enviar tus cambios locales al repositorio remoto.

**Fetch:** Descargar los cambios del repositorio remoto a tu local, sin unirlos automáticamente.

**Clone:** Copiar un repositorio remoto completo a tu computadora local.

**Remote (Remoto):** Dirección de un repositorio en línea (por defecto se llama "origin").

**Fork (Bifurcación):** Copia personal de un repositorio para experimentar sin dañar el original.

**Pull request (Solicitud de extracción):** Propuesta formal para que tus cambios en un fork se incluyan en el repositorio original.

**Conflictos de merge:** Problemas cuando Git no puede unir cambios automáticamente porque se editaron las mismas partes de un archivo. Requiere solución manual.