GA7-220501096-AA1-EV03 IDENTIFICA HERRAMIENTAS DE VERSIONAMIENTO

APRENDICES
KATHERINE VALENCIA ALZATE
XABI DANIEL PEDREROS ANDAPIÑA
SEBASTIAN VANEGAS
DANEYI CATERIN ESPINOSA

INSTRUCTOR
JHON JAIRO DONATO

TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE
FICHA 2834889

SERVICIO NACIONAL DE APRENDIZAJE – SENA

2024

INTRODUCCIÓN

Git es una herramienta de control de versiones distribuido que permite gestionar los cambios en el código de forma eficiente y colaborativa. Al trabajar con Git, es fundamental entender la diferencia entre el repositorio local**,** que se encuentra en el equipo del usuario, y el repositorio remoto**,** que se aloja en un servidor y permite la colaboración en equipo.

En Git local, el control de versiones se maneja en el equipo propio, proporcionando rapidez y privacidad hasta que los cambios se sincronizan. En cambio, Git remoto facilita el trabajo en equipo al hacer que los cambios sean accesibles para otros colaboradores. Ambos entornos ofrecen comandos específicos para la gestión de versiones, cada uno enfocado en operaciones locales o en sincronización con el servidor remoto.

La siguiente tabla detalla las diferencias clave entre el sistema de Git local y Git remoto, junto con los comandos básicos que permiten operar en cada uno de estos entornos. Esta información es esencial para aprovechar al máximo las funcionalidades de Git en proyectos individuales y colaborativos.

OBJETIVOS

● **Comprender la estructura de Git**: Explicar las diferencias entre el repositorio local y el repositorio remoto, y cómo cada uno contribuye al control de versiones en un proyecto.

● **Identificar los beneficios de Git local y Git remoto**: Presentar las ventajas de utilizar Git local para trabajo individual y Git remoto para facilitar la colaboración y la sincronización de cambios en un equipo.

● **Conocer los comandos básicos de Git**: Familiarizar a los asistentes con los comandos esenciales para gestionar versiones tanto en el repositorio local como en el remoto, facilitando el flujo de trabajo en Git.

● **Desarrollar habilidades prácticas en Git**: Proporcionar una guía clara que permita a los usuarios aplicar los comandos en entornos locales y remotos, mejorando su productividad en proyectos de desarrollo.

● **Fomentar el uso de buenas prácticas en control de versiones:** Promover una comprensión completa del sistema de control de versiones de Git, enfatizando la importancia de sincronizar y coordinar cambios entre el repositorio local y el remoto para evitar conflictos y mantener la integridad del proyecto.

ASPECTO	GIT LOCAL	GIT REMOTO
Ubicación de los datos	Almacenados en el equipo local del usuario, en una carpeta .git	Almacenados en un servidor accesible por red o internet
Acceso	Solo accesible en la máquina del usuario	Accesible para múltiples usuarios, según los permisos
Velocidad	Rápido, ya que no depende de la conexión a internet	Puede ser más lento debido a la transferencia de datos
Privacidad	Cambios y commits son privados hasta que se sincronizan	Los cambios pueden ser visibles para otros colaboradores
Colaboracion	Generalmente individual, aunque se puede compartir manualmente	Facilita la colaboración en equipo
Historia del proyecto	Almacena la historia de cambios en la copia local	Permite tener una copia completa de la historia de cambios en varios servidores

COMANDOS BASICOS DE GIT LOCAL

COMANDO	DESCRIPCION
git init	Inicializa un nuevo repositorio Git en el directorio actual.
git add <archivo>	Añade el archivo especificado al área de staging para preparar el commit.
git commit -m "<mensaje>"	Guarda los cambios en el historial de versiones, con un mensaje descriptivo.
git status	

	Muestra el estado de los archivos, indicando cuáles están en staging o sin cambios.
git log	Muestra el historial de commits en el repositorio local.
git diff	Compara cambios entre el área de trabajo y el área de staging.

COMANDOS BASICOS DE GIT REMOTO

COMANDO	DESCRIPCION
git remote add <nombre> <url>	Agrega un nuevo repositorio remoto con un nombre específico.
git push <remoto> <rama>	Sube los cambios de una rama al repositorio remoto especificado.
git fetch <remoto>	Descarga los cambios del repositorio remoto, pero no los integra en el repositorio local.
git pull <remoto> <rama>	Descarga e integra los cambios de la rama del repositorio remoto a la rama actual del repositorio local.
git clone <url>	Crea una copia del repositorio remoto en el equipo local.
git remote -v	Muestra la lista de repositorios remotos con sus URLs correspondientes.