GA7-220501096-AA1-EV01 IDENTIFICACION DE HERRAMIENTAS DE VERSIONAMIENTO

Robert Alexi Cachay Neme

SENA regional Casanare

ADSO

INTRODUCCION

En el contexto del desarrollo de software, la integración continua es una práctica esencial que permite a los equipos de desarrollo integrar cambios en el código de manera frecuente y automática. Git, como sistema de control de versiones distribuido, juega un papel crucial en esta práctica, facilitando tanto el control de versiones local como remoto. Este documento tiene como objetivo explorar las diferencias y características entre Git local y Git remoto, proporcionando una comprensión clara de cómo cada uno contribuye al flujo de trabajo de integración continua.

OBJETIVO

El objetivo de este documento es comparar y contrastar los sistemas de control de versiones Git local y Git remoto, destacando sus diferencias y características. Además, se proporcionarán los comandos básicos utilizados en ambos contextos para facilitar su uso y comprensión.

1. Tomando como referencia el componente formativo “Integración continua”, realice una tabla con las diferencias entre el sistema de control de versiona miento git local y git remoto.

|  |  |  |
| --- | --- | --- |
| Característica | Git Local | Git Remoto |
| Almacenamiento | Almacena los cambios en el repositorio local | Almacena los cambios en un servidor remoto |
| Acceso | Solo accesible desde la máquina local | Accesible desde cualquier ubicación con acceso |
| Colaboración | Limitada a un solo usuario | Permite colaboración entre múltiples usuarios |
| Seguridad | Depende de la seguridad del dispositivo local | Depende de la seguridad del servidor remoto |
| Rendimiento | Operaciones rápidas debido a la falta de red | Puede ser más lento debido a la latencia de red |
| Comandos Clave | git commit, git log, git diff | git push, git pull,git fetch |

1. Realice una tabla con la descripción de los comandos básicos de git remoto y git local

|  |  |
| --- | --- |
| **Comando** | **Descripción** |
| Comandos de Git Local | |
| git init | Inicializa un nuevo repositorio Git en el directorio actual. |
| git add <archivo> | Añade archivos al área de preparación (staging area). |
| git commit -m "mensaje" | Guarda los cambios en el repositorio local con un mensaje descriptivo. |
| git status | Muestra el estado de los archivos en el directorio de trabajo y el área de preparación. |
| git log | Muestra el historial de commits. |
| git diff | Muestra las diferencias entre los cambios realizados. |
| git branch | Lista, crea o elimina ramas. |
| git checkout <rama> | Cambia a una rama específica. |
| git merge <rama> | Fusiona una rama con la rama actual. |
| git stash | Guarda temporalmente los cambios no confirmados. |
| git stash pop | Aplica los cambios guardados en el stash. |
| Comandos de Git Remoto | |
| git remote add origin <URL> | Añade un repositorio remoto. |
| git push | Envía los cambios locales al repositorio remoto. |
| git pull | Trae los cambios del repositorio remoto al local y los fusiona. |
| git fetch | Descarga los cambios del repositorio remoto sin fusionarlos. |
| git clone <URL> | Clona un repositorio remoto a la máquina local. |
| git remote -v | Muestra los repositorios remotos configurados. |
| git push origin <rama> | Envía una rama específica al repositorio remoto. |
| git pull origin <rama> | Trae y fusiona una rama específica del repositorio remoto. |
| git remote remove <nombre> | Elimina un repositorio remoto. |
| git push --tags | Envía todas las etiquetas locales al repositorio remoto. |