

GA7-220501096-AA1-EV03

IDENTIFICA HERRAMIENTAS DE VERSIONAMIENTO

ONALDO POLO PACHECO

APRENDIZ

CRISTIAN CAMILO ARIAS TIBAQUIRA

INSTRUCTOR

CENTRO DE FORMACIÓN AUTOMATIZACIÓN INDUSTRIAL

SENA REGIONAL CALDAS

TECNÓLOGO EN ANÁLISIS Y DESARROLLO DEL SOFTWARE

FICHA: 2721448

INTRODUCCIÓN

El control de versiones es fundamental para gestionar el código fuente de manera eficiente. Git es un sistema de control de versiones distribuido, se ha convertido en la herramienta estándar de la industria debido a la flexibilidad que nos brinda.

En este informe conoceremos en que se diferencia el sistema de control de versiones git de los demás sistemas de control de versiones, las características y diferencias de git local y remoto y algunos de los códigos básicos de git local y remoto.

SISTEMA DE CONTROL DE VERSIONES

Git, a diferencia de otros sistemas de control de versiones, posee un sistema de almacenamiento de información en el que cada versión posee una copia en miniatura, nos ofrece la posibilidad de trabajar aun cuando no tenemos conectividad con el servidor central, ya que localmente contamos con toda la información necesaria.

Debido a que git solo añade información a la base de datos con cualquier acción realizada, es muy difícil llegar a un estado donde este sistema de control de versiones no pueda hacer un proceso de recuperación o que se elimine la información una vez exista una confirmación de los cambios.

CARACTERÍSTICAS Y DIFERENCIAS DE GIT LOCAL Y REMOTO

CARACTERÍSTICAS	GIT LOCAL	GIT REMOTO
Almacenamiento	Máquina del usuario	Servidor remoto
Acceso	Solo local	A través de internet
Colaboración	Solo usuario o equipo local	Múltiples usuarios
Respaldo	Depende del hacer copias de seguridad	Mayor seguridad en respaldos remotos
Historial	Historial de los cambios solo localmente	Historial de cambios accesible por todos los usuarios
Compartir código	Requiere métodos externos	Facilita compartir código entre el equipo
Sincronización	No se sincroniza automáticamente con otros repositorios	Se sincroniza automáticamente con el repositorio remoto
Acceso desde cualquier lugar	No se puede acceder desde otras ubicaciones	Acceso desde cualquier lugar con conexión a internet

COMANDOS BASICOS

Comando	Descripción	Git Local	Git Remoto
git init	Inicializa un nuevo repositorio Git en el directorio actual.	Sí	No (se utiliza en el directorio local)
git clone	Clona un repositorio Git existente en un nuevo directorio.	No (se utiliza en el directorio local)	Sí
git add	Agrega cambios al área de preparación.	Sí	No (se utiliza en el directorio local)
git commit	Registra los cambios en el repositorio.	Sí	No (se utiliza en el directorio local)
git push	Envía los cambios locales al repositorio remoto.	No (se utiliza en el directorio local)	Sí
git pull	Obtiene y fusiona los cambios del repositorio remoto en el directorio de trabajo local.	No (se utiliza en el directorio local)	Sí
git fetch	Descarga los cambios del repositorio remoto a tu repositorio local, pero no los fusiona con tu trabajo actual.	No (se utiliza en el directorio local)	Sí
git remote add	Agrega un nuevo repositorio remoto.	No (se utiliza en el directorio local)	Sí
git remote -v	Muestra los repositorios remotos configurados.	No (se utiliza en el directorio local)	Sí
git remote remove	Elimina un repositorio remoto.	No (se utiliza en el directorio local)	Sí
git status	Muestra el estado de los archivos en el directorio de trabajo.	Sí	Sí (también muestra información sobre el estado del repositorio remoto)
git log	Muestra el historial de confirmaciones.	Sí	Sí (también puede mostrar el historial remoto)

CONCLUSIÓN

El uso de git, tanto en entornos locales como remotos, es esencial para el éxito de los proyectos, incluyendo crear una tienda deportiva en línea. Al comprender los comandos básicos de git, podremos gestionar de manera eficiente el ciclo de vida del software, podremos colaborar con otros del equipo y mantener un historial de los cambios realizados.

En pocas palabras, el uso de git es fundamental en nuestros proyectos, ya que git es una herramienta indispensable para cualquier proyecto de software moderno, dominarlo es fundamental para alcanzar el éxito en nuestros proyectos.